2006

Multiple View Geometry For Video Analysis And Post-production

Xiaochun Cao
University of Central Florida

MULTIPLE VIEW GEOMETRY FOR VIDEO ANALYSIS AND POST-PRODUCTION

by

XIAOCHUN CAO
B.E. Beijing University of Aeronautics and Astronautics, 1999
M.E. Beijing University of Aeronautics and Astronautics, 2002

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2006

Major Professor:
Dr. Hassan Foroosh

ABSTRACT

Multiple view geometry is the foundation of an important class of computer vision techniques for simultaneous recovery of camera motion and scene structure from a set of images. There are numerous important applications in this area. Examples include video post-production, scene reconstruction, registration, surveillance, tracking, and segmentation. In video post-production, which is the topic being addressed in this dissertation, computer analysis of the motion of the camera can replace the currently used manual methods for correctly aligning an artificially inserted object in a scene. However, existing single view methods typically require multiple vanishing points, and therefore would fail when only one vanishing point is available. In addition, current multiple view techniques, making use of either epipolar geometry or trifocal tensor, do not exploit fully the properties of constant or known camera motion. Finally, there does not exist a general solution to the problem of synchronization of N video sequences of distinct general scenes captured by cameras undergoing similar ego-motions, which is the necessary step for video post-production among different input videos.

This dissertation proposes several advancements that overcome these limitations. These advancements are used to develop an efficient framework for video analysis and post-production in multiple cameras. In the first part of the dissertation, the novel inter-image constraints are introduced that are particularly useful for scenes where minimal information is available. This result

extends the current state-of-the-art in single view geometry techniques to situations where only one vanishing point is available. The property of constant or known camera motion is also described in this dissertation for applications such as calibration of a network of cameras in video surveillance systems, and Euclidean reconstruction from turn-table image sequences in the presence of zoom and focus. We then propose a new framework for the estimation and alignment of camera motions, including both simple (panning, tracking and zooming) and complex (e.g. hand-held) camera motions. Accuracy of these results is demonstrated by applying our approach to video post-production applications such as video cut-and-paste and shadow synthesis. As realistic image-based rendering problems, these applications require extreme accuracy in the estimation of camera geometry, the position and the orientation of the light source, and the photometric properties of the resulting cast shadows. In each case, the theoretical results are fully supported and illustrated by both numerical simulations and thorough experimentation on real data.

Dedicated to my wonderful wife *Zhe Tian*, and my caring parents.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my advisor, Dr. Hassan Foroosh, who has made this dissertation possible. I am indebted to him for his guidance, inspiration, encouragement and understanding throughout my graduate studies. I have learned a lot from him about doing research and presenting results.

I would also like to thank Dr. Michael Georgiopoulos, Dr. Mark Heinrich, Dr. Charles Hughes and Dr. Annie Wu for serving as my committee members, and for their invaluable comments and suggestions.

This dissertation is built upon the work that appeared in several papers, which I coauthored with my advisor and collaborators, Dr. Jiangjian Xiao, Yuping Shen, Murat Balci, Imran Junejo, Fei Lu, and Yun Zhai who contributed to this work in many different ways. I would like to especially thank Dr. Mubarak Shah with whom I worked for four semesters during my Ph.D program, and who supported me during that time.

I wish to thank the truly amazing people of the Computational Imaging Lab, who made this lab such stimulating and pleasant environment to carry out research.

Most importantly, I want to thank my wife, Zhe Tian, for her endless love and confidence in me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Image acquisition is the process of formation of a two-dimensional representation of a three-dimensional world. Conversely, an important goal in computer vision is to recover the camera geometry and deduce the 3D structure of the scene that appears in the images simultaneously. Although the traditional *shape from X* approaches, such as shape from shading [73, 207], shape from texture [177, 116], shape from specularity [85, 126] and shape from defocus [130, 36], are extremely useful for specific applications, none of them is as flexible as, or comparatively as accurate as stereo or multi-view geometry-based methods [49, 55, 75]. One important reward of the multiple view geometry based 3D reconstruction techniques is the video post-production, the problem addressed in this dissertation. While video post-production includes a variety of phases such as video editing, audio mixing, special effects and distributing the finished product, this dissertation mainly tackles the problem of generating visual special effects using computers. Automatic reconstruction techniques have recently become widely used in the film industry as a means for adding synthetic objects in real video sequences, where computer analysis of the motion of the camera and the structure of the scene is replacing the previously used manual methods for correctly modeling and aligning the artificially inserted objects.

1.1 Motivation

The problem of uncalibrated video analysis and post-production is tackled in this thesis. In designing a system for post-processing videos, different tradeoffs are proposed by each application. Cost, accuracy, ease of use and robustness are the main parameters to be considered. In film and TV production, cranes are mostly used to control camera movement, while blue screen techniques are adopted to create special effects. Computer vision researchers aim to reduce the cost, and simplify these processes, possibly by trading off some amount of accuracy, and by maximizing the use of information that is available in the images or videos. Toward the goal of post-producing videos using standard home and office equipment, such as a PC and a video camera, five related technical challenges have been addressed in this thesis:

1. camera calibration using scene properties (Chapter 4);
2. self-calibration using camera motion (Chapter 5);
3. camera motion analysis and video synchronization (Chapter 6)
4. video post-production (Chapter 7);
5. the estimation of scene photometry and shadow synthesis (Chapter 8).

Each of these techniques has a long history in computer vision, computer graphics or multimedia, with remarkably rich literature (an extensive bibliography to the subjects can be found in the corresponding chapters). However, it is rare to find works that efficiently handle both the analysis and the post-production for general videos in the same framework.

1.1.1 Video Post-production

In the film or TV industry, video post-production is the main step after shooting film or video, and recording audio. Basically, video post-production includes gathering visual and audio assets (e.g. "clips"); developing visual effects, titles and graphics; editing the production, and distributing the finished product. In this work, we mainly tackle the computer-based problem of digital visual effects. The general term used in the industry is CG, short for computer-generated. For example, when talking to the artists you will hear them say things like, "That entire scene is CG," or "Those are all CG soldiers," or "The actors are real, but everything else is CG." Computer-generated effects make imaginary characters like Godzilla possible, and they also create almost every effect that used to be done using models. The advantages of CG effects are their realism, flexibility and relatively low cost (compared to the alternatives).

The origins of video post-production techniques can be traced back to photography and cinematography. One early example is the massive print, "The Two Ways of Life", created by the photographer Oscar Rejlander in 1857 by selectively combining 32 different negatives [19]. As argued by [31], nearly all modern movies utilize cut and paste operations in their production, e.g. "Jurassic Park", "The Matrix", and "The Lord of the Rings". As of April 2004, six of the ten best-selling movies ever released also won the "Best Visual Effects" Academy Awards. In addition to being put to use for visual effects, digital video post-productions are used in much of today's media, including magazines, 2D arts and graphics, television, advertising, and multimedia title development.

Although they were developed several decades ago, most existing video post-production techniques are still fairly manual intensive, and often limited to solving a restricted version of the general problem. One major limitation in most of the past works, e.g. [22, 3, 102], is that they assume the cameras capturing both the source and target scenes are fixed or have a known 2D translation or scaling. Therefore, a direct alpha blending [144] is sufficient to composite the foreground objects into the target frames. However, camera motion is known as one of the most important camera techniques used in the production of television and movie, since there are certain common conventions that convey meaning through particular camera techniques. Therefore, for the video post-production to be compelling, it is important to analyze the motion of the cameras in the video shots, and choose and align source and target shots which have similar motions. In addition to the correctness of the geometry, the other important factor in video post-production is the photometric consistency of the synthesized objects with respect to the existing objects in the original scene.

1.1.2 Geometry Based Video Analysis

The analysis of video data generally targets the identification of relevant objects or regions in a scene (*segmentation*), and the extraction of associated descriptive characteristics for each object/region and for the complete scene (*feature extraction*). The video analysis herein concentrates on low level feature extraction, the camera motion characterization, and the temporal video alignment, which are necessary components to cut and paste objects among videos captured by moving cameras.

The recognition of video shots in which camera is static, zooming, or rotating, has been achieved using rather dedicated methods [13, 122, 47]. Usually, these methods rely on the exploitation of motion vectors issued from block-matching techniques, or on the search for specific distributions of motion vectors or a few global representative motion parameters [204]. However, one of the main shortcomings of these approaches is that they are generally not resilient to the presence of mobile objects of significant size. In addition, it is difficult for these methods to distinguish pure rotation shots from pure translation shots, e.g. the panning and the side-way tracking shots, since they mostly use inter-image planar homography, which is valid only when the camera positions are fixed or the scene is almost planar.

The problem of temporally aligning video sequences has become an active area in computer vision community since Stein's first method [162]. More recent methods [170, 34, 166, 185, 96] tackle the problem of automatic video synchronization for independently moving cameras and overlapping dynamic scenes. The alignment of non-overlapping sequences was first addressed by Caspi and Irani [33] based on the assumption that the two sequences are captured by a stereo rig. Nevertheless, previous efforts [33, 42, 140] on temporal alignment of non-overlapping sequences typically utilize the inter-sequence relationship and hence inherently involve two sequences. For many video sequences, the cameras filming the scenes are moving in a random fashion, e.g. a hand-held shot. To post-process such video shots, it is necessary to accurately recover the motions of the cameras that capture the videos, which is detailed in the next section.

1.1.3 Complex Camera Motion Recovery

For video shots where the cameras are moving freely, the most important requirement for video post-production, e.g. realistic insertion of an artificial object in the sequence, is to compute the correct motion of the camera. This is not as simple as the method for shots captured by purely rotating or zooming cameras as described in the above Section 1.1.2. Unless the camera motion is correctly determined [75], it is impossible to generate the correct sequence of views of the inserted object or re-render the objects matted from the source videos in a way that will appear geometrically consistent with the target video. Generally, it is only the motion of the camera that is important here. One does not need to reconstruct the scene, since it is already present in the existing video, and novel views of the scene visible in the video are not required. The only requirement is to be able to generate correct perspective views of the inserted object or re-render the objects matted from the source videos.

Except for when the inserted object and the cameras are known in the same coordinate frame [161], the generated views of the inserted object will be seen distorted with respect to the perceived structure of the scene in the target video. Since the geometry of the object to be inserted is in practice Euclidean, it is essential to compute the motion of the camera also in a Euclidean frame, or at least metric frame in the looser case. Therefore, it is not enough to know merely the projective motion of the camera, which can be obtained from the computed fundamental matrices or trifocal tensors [75]. Traditionally, camera internal parameters and the external parameters (motion) can be calibrated using known objects, e.g. a 3D square grid [174], a 2D square grid [200], or scene

properties such as the vanishing points [44, 25, 105] and the circular points [186, 45, 23]. Recently, self-calibration methods [172, 134, 4, 156] are proposed to avoid the onerous task of calibrating cameras using special calibration objects, and thus make it possible to calibrate a camera directly from an image sequence despite unknown motion and changes in some of the internal parameters. However, the current state-of-the-art vanishing point based techniques are not able to handle the situations where only one vanishing point is available, and the existing self-calibration methods do not fully exploit the properties of constant camera motions, which is popular in the real world.

1.1.4 Photometric Information Recovery

Video post-production requires not only the geometric correctness but also the photometric consistency. The photometric information here is mainly the light source location and illumination properties required to match the color characteristics of the synthesized shadows of the inserted objects with those of the original scene. Shadows provide important visual cues for depth, shape, contact, movement, and lighting in our perception of the world [90].

In previous efforts, shadows for video post-production are typically either created manually or composited using matted shadows from the source scenes. The former approach is commonly called “faux shadow” in the film industry [187]. In this technique, artists use the foreground object’s alpha matte to create its shadow manually. Consequently, the geometric accuracy of the “faux shadow” highly depends on the experience of the artists, and the color characteristics of the

"faux shadow" are interactively adjusted by the compositor. In the recent work by [132], a semi-automatic method for creating shadow mattes in character animation is presented. Their system creates shadow mattes based on hand-drawn characters, given high-level guidance from the user regarding the depths of various objects. The method employs a scheme for "inflating" a 3D figure based on hand-drawn art. It provides simple tools for adjusting object depths, coupled with an intuitive interface by which the user specifies object shapes and the relative positions in a scene. Their system obviates the tedium of drawing shadow mattes by hand, and provides control over complex shadows falling over interesting shapes. However, the method has difficulties in obtaining models in video post-production operations among given real videos.

The second kind of approach is to extract shadows from the source scenes using luma keying or alpha matting [144, 30], which are efficient when both the target and source scenes are accessible. The accessibility here means that the target scene setup is controllable and known, i.e. we are able to obtain the information about the camera and light source. However, these methods would not simply apply to general cases where the source and target video shots have different camera motions, since the relative orientations between the camera and the light source might not be the same in the source and the target video. In a more recent work [123], a semi-automatic 2D shadow synthesis method is proposed, in which the video object plane and shadow contours are approximated from an existing reference one. However, this approach still involves nontrivial interactive operations and the effects of the generated shadows depend highly on how much these strict assumptions are satisfied.

1.2 The Novel Framework

The proposed novel framework, herein, tackles the problem of video analysis and post-production in a divide-and-conquer fashion. Based on the explicit geometry-based analysis of the camera motion of a shot, we classify and align that shot into either simple camera motion shot, e.g. panning, tracking and zooming, or complex camera motion shot, e.g. hand-held shots, and then solve different cases using dedicated methods.

For simple camera motions, such as zooming, pure rotation and pure translation, the inter-frame homographies and fundamental matrices have special forms and hence additional properties, which can be used to classify and align shots. In the case of pure translational camera motion, we compute the relative translational magnitude from the slices cut from the three dimensional data volume along the epipolar lines. For cameras with fixed locations undergoing pure rotation, we compute the rotation angles directly from the inter-frame planar homography by eigenvalue decomposition. The frames captured by cameras undergoing only zooming effects can be related by inter-image planar homography, from which it is easy to extract the zoom factor with respect to a reference image based on the assumption of a simplified pin-hole camera model.

To tackle a relatively more general problem, i.e. the quantification and alignment of N video sequences of distinct general scenes captured by cameras undergoing similar ego-motion, we observe that similar camera displacements result in the same relative inter-frame translation and orientation, i.e. the same essential matrices, between synchronized frame pairs, and also that the equality of the essential matrix is reflected in the uncalibrated fundamental matrix in that its upper-left 2×2

elements remain constant up to an unknown scale. Note that two camera motions are similar when the camera locations and poses in all corresponding time slots are related by a common 3D similarity transformation. Therefore, for each frame, we can obtain a homogeneous four-dimensional feature vector characterizing the camera ego-motion relative to a reference frame. The relative translational magnitude (for pure translational camera motion), rotation angles (for pure rotating camera motion), zoom factor (pure zooming camera motion) and the four-dimensional feature vector (for general camera motion) are used to temporally align video shots which undergo the similar camera motion.

For the shots with complex or combined camera motion, e.g hand-held shots, it is not always likely to have another shot which has similar camera motions. In these cases, to ensure that the virtual pasted objects are consistent with the existing objects in the target shots, the proposed framework first explicitly estimates the poses of the cameras that captured both the source and target scenes using camera calibration techniques. Then, the videos are aligned both temporally and spatially along the computed camera trajectories in 3D space by minimizing the global differences between the source and transformed target viewing directions. Finally, the depth information of the foreground object in the source shot is recovered, which is used for the foreground layers to be re-rendered and blended into the corresponding target frames.

After the alignment among the source and target videos, the objects are cut from the source video using the technique described in [194, 191], which combines graph-cuts based motion layer segmentation and poisson alpha matting. Finally, to increase the realism of the composited videos, the photometric properties, such as shadows of the inserted objects, are synthesized based on the

geometric and photometric constraints extracted from the target images or videos. The geometric constraints are mostly obtained by camera calibration methods. We increase the accessibility of calibration objects by introducing shadows, symmetric and 1D objects, and make use of constant camera motion for self-calibration. The self-calibration can also be used to recover the 3D model of an object, which is useful in video augmented reality applications. In the case when camera calibration is not possible, we also develop a method for shadow synthesis of planar or distant object using planar homography.

Overall, the new framework, compared to the current state-of-the-art techniques, has the following advantages. First, it is able to post-process, e.g. cut and paste, objects among general video shots. Second, the framework does not require the relationship between the dominant light source, the reference plane, and the camera to be the same in the source and target scenes. Third, the cameras capturing the source and the target videos are not necessarily fixed or move simply along a simple linear path. Finally, no 3D knowledge about either the source or the target scenes are required in the new framework.

1.3 Contributions

This thesis improves on the state of the art on various aspects of computer vision, and understanding of photographs and visual art:

- The proposed inter-image constraints extend the power of the state-of-the-art projective geometry techniques to situations where only one vanishing point is available, and therefore are especially useful for scenes where only minimal information is available, e.g. an isosceles trapezoid obtained from symmetric objects.

- Three common objects (symmetric objects, 1D objects, and vertical objects and their shadows) are proposed for camera calibration, which have the benefit of increasing the accessibility of the calibration objects.

- Constant or known camera displacement is used for self-calibration and is applied to Euclidean reconstruction from turn-table sequences in the presence of zoom and focus. These results can also be used for model-based post-production applications, e.g. augmented reality.

- A new video alignment method that can deal with video sequences of general distinct scenes captured by arbitrarily, both generally and specially, moving cameras. The proposed approach is a 1-sequence process and computes the camera ego-motion for each sequence separately. Besides its efficiency allowing a combination-free implementation, the algorithm, as a 2D solution, does not involve scene reconstruction or 3D recovery of the camera trajectories.

- An efficient framework is developed for video post-production applications such as cut-and-paste. Different from previous work, this framework is able to handle cases where the

source and the target videos have nontrivial unknown camera motions, and requires no 3D knowledge about either the source or the target scenes.

- A pragmatic framework for rendering shadows composited into target views based on geometric and photometric analysis is proposed. This framework is flexible in that various techniques suitable for different scenes can be easily integrated in it.

1.4 Overview of the Dissertation

Chapter 2. Before discussing the proposed framework, we begin with a literature survey of the most relevant research conducted in the areas of: camera calibration using objects, self-calibration, video analysis and post-production.

Chapter 3. This chapter provides a brief introduction to single and multiple view geometry used elsewhere in this dissertation. It first introduces the pinhole camera model, and the perspective mappings including planar homography and planar homology. It then describes the calibration methods using orthogonal constraints, i.e. the mutually orthogonal vanishing points and the image of the absolute conic. Finally, the epipolar geometry in two views is also discussed.

Chapter 4. This chapter describes in detail one of the main contributions of this work, which is the derivation of the new inter-image constraints. These constraints are utilized to develop al-

gorithms for camera calibration making use of three calibration objects (symmetric objects, 1D objects, and vertical objects and their parallel shadows on the planar ground plane). The implementation details and degenerate configurations of the algorithm are also shown in this chapter. Finally, we demonstrate the applications to reconstruction of partially viewed symmetric objects and image based metrology.

Chapter 5. This chapter proposes the new constraint using constant or known camera displacement for self-calibration. It explores some equality properties of the essential matrix between neighboring frame pairs, which are used to develop a linear algorithm for the computation of focal lengths given the inter-frame fundamental matrices and knowledge of remaining camera internal parameters. This constraint is applied to self-calibration from turn-table sequences in the presence of zoom and focus. This method is also extended for Euclidean reconstruction of objects rotating on a turn-table, which can then be used for instance in augmented reality applications.

Chapter 6. This chapter addresses the problem of synchronizing video sequences of distinct scenes captured by cameras undergoing similar motions. For the general motion and 3D scene, the camera ego-motions are featured by parameters obtained from the fundamental matrices. In the case of special motion, e.g. pure translation and rotation, relative translational magnitude or rotation angles are used as motion features. These extracted features are invariant to the camera internal parameters, and can be computed without recovering camera trajectories along the image sequences. Experimental results show that our method can accurately synchronize sequences even

when they have dynamic timeline maps, and the scenes are totally different and have dense depths.

Chapter 7. This chapter describes a video-based framework for a video post-production operation, i.e. to pull the alpha mattes of rigid or approximately rigid 3D objects from one or more source videos, and then use them to augment a target video of a different scene in a geometrically correct fashion. This framework builds upon techniques in camera pose estimation, 3D spatiotemporal video alignment, depth recovery, key-frame editing, nature video matting, and image-based rendering. Experimental results on various content types, e.g. TV programs, home videos and feature films, and different camera motions are reported to validate the proposed framework.

Chapter 8. This chapter mainly tackles another problem related to video post-production, the shadow and reflection synthesis. To synthesize shadows and reflection of the inserted objects, the framework efficiently utilizes the geometric and photometric constraints extracted from the target images or videos. In addition to strong geometric constraints obtainable from camera calibration, the planar homology constraint is introduced for cases where camera calibration is not possible. This chapter also demonstrates how to constrain the synthesized shadows and reflections to be photometrically consistent with those in the original target views. Finally, to show the accuracy and the applications of the proposed method, this chapter presents the results for a variety of target scenes, including footage from commercial Hollywood movies and 3D video games.

Chapter 9. The conclusion presents a summary of the work and points out the directions for future research on the topics addressed in this dissertation.

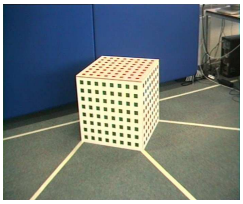
CHAPTER 2

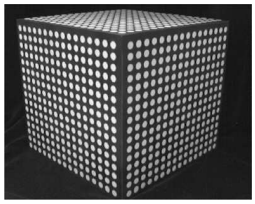
RELATED WORK

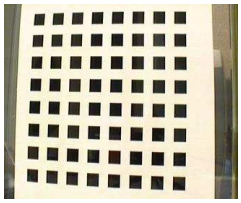
This chapter presents a survey of the most significant work in the field of camera calibration, three dimensional Euclidean reconstruction from two-dimensional images, video analysis, and video post-production.

2.1 Camera Calibration

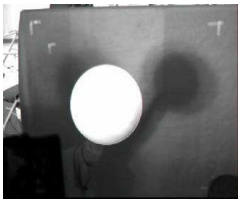
There has been much work on camera calibration, both in photogrammetry and computer vision. Existing methods generally involve some trade-off between automation and accuracy, and can be broadly classified into two categories. The first category includes methods that rely on the scene properties such as known 3D coordinates of a calibration grid, which is the topic of this section. The second category consists of methods, generally referred to as self-calibration or auto-calibration approaches, that aim to compute a metric reconstruction from multiple uncalibrated images and avoid the onerous task of calibrating cameras using special calibration objects.


(a) 3D square grid

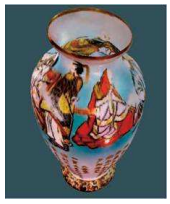

(b) 3D circular grid


(c) 2D square grid


(d) Architectural buildings


(e) Balls or spheres


(f) Coplanar circles


(g) SOR

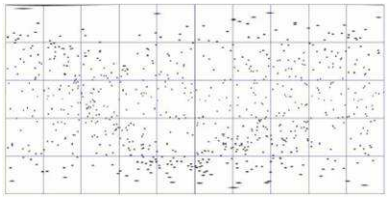

(h) Fixed stars in the night sky

Figure 2.1: A partial summary list of previous camera calibration objects. Note that this figure is by no means complete and only representative work is listed.

2.1.1 Camera Calibration Using Scene Properties

Traditional methods in the first category use a calibration object with a fixed 3D geometry, e.g. 3D square grid [174] (see Figure 2.1.a) and 3D circular grid [69] (see Figure 2.1.b). Recently, more flexible plane-based calibration methods [200, 153, 58, 107] have been proposed that use the orthonormal properties of the rotation matrix, which represents the relative rotation between the 3D world and the camera coordinate systems. Zhang [200] has originally shown that it is possible to calibrate a camera using a planar pattern (see Figure 2.1.c) observed at a few different orientations, and has obtained very accurate results. Sturm and Maybank [153] extended this technique, and obtained the solution for the case when the camera intrinsic parameters can vary. They also describe all the singularities, naming the parameters that cannot be estimated, for some

minimal cases. Gurdjos et al. [58] formulated the plane-based calibration problem into a more intuitive geometric framework, according to Poncelet's theorem, which makes it possible to give a Euclidean interpretation to the plane-based calibration techniques.

Some recent methods using calibration rigs rely also on non-planar objects. For instance, in [2, 176, 210], balls or spheres (see Figure 2.1.e) are used to solve for the camera intrinsic parameters and the locations of the spheres. Architectural buildings [44, 25, 105] (see Figure 2.1.d), surfaces of revolution (SOR) [186, 23] (see Figure 2.1.g), coplanar circles [111, 81, 45] (see Figure 2.1.f), non textured Lambertian surface [94], and fixed stars in the night sky [84] (see Figure 2.1.h) are used as alternative calibration objects. the method is not able to extract highly accurate camera parameters from real images. Zhang [202] has also presented a method for camera calibration using an 1D object pivoting on a fixed point - thereby filling in the missing dimension in the use of calibration rigs. One-dimensional objects are also used in [27, 26, 184] for camera calibration. Figure 2.1 shows some previous camera calibration objects. Typically, methods falling into this category provide very accurate results. In some applications, however, it might not be possible to extract camera information off-line by using calibration objects due to the inaccessibility of the camera.

2.1.2 Self-Calibration

Self-calibration differs from conventional calibration where the camera internal parameters are determined from the image of a known calibration grid or properties of the scene. The prefix *self-* is added as soon as the world's Euclidean structure is unknown, which can be seen as a case of "0D" calibration. In self-calibration the metric properties of the cameras are determined directly from constraints on the internal and/or external parameters.

The first self-calibration method, originally introduced into computer vision by [56], involves the use of the Kruppa equations. The Kruppa equations are two-view constraints that require only the fundamental matrix to be known, and consist of two independent quadratic equations in the elements of the dual of the absolute conic. Luong and Faugeras [139] have shown that the Kruppa equations are equivalent to the Trivedi constraints [171] and the Huang-Faugeras constraints [70, 65], although the equivalence does not mean that they will produce the same results when used in self-calibration algorithms. Algorithms for computing the focal lengths of two cameras given the corresponding fundamental matrix and knowledge of the remaining intrinsic parameters are provided by Hartley [65] and Bougnoux [16]. Mendonça [110] generalized the results in [65, 16] for an arbitrary number of cameras and introduced a built-in method for the detection of critical motions for each pair of images in the sequence. Thorough analyses of critical motions which would result in ambiguous solutions by Kruppa-based methods are described in [163] and [93].

An alternative direct method for self-calibration was introduced by Triggs [172], which estimates the absolute dual quadric over many views. The basic idea is to transfer a constraint on the

dual image of absolute conic to a constraint on the absolute dual quadric, and hence determine the matrix representing the absolute dual quadric, from which a rectifying 3D homography can be decomposed that transforms from projective to metric reconstruction. Heyden and Astrom [62] showed that metric reconstruction was possible knowing only skew and aspect ratio, and Pollefeys et. al. [134] and Heyden and Astrom [63] showed that zero skew alone was sufficient. In addition, Pollefeys [134] developed a practical method for self-calibration of multiple cameras with varying intrinsic parameters, and showed results for real sequences.

Special motions can also be used for self-calibration. Agapito et al. [4] and Seo and Hong[148] solved the self-calibration of a rotating and zooming camera using the infinite homography constraint. Before their work, Hartley [66] solved the special case where the camera's internal parameters remain constant throughout the sequence. Frahm and Koch [54] showed it was also possible to solve the problem of generally moving camera with varying intrinsics but known rotation information. Triggs [173] provided a solution for self-calibration from scene planes when the internal parameters are constant, and Zisserman et. al. [196] presented a method for self-calibration of a stereo rig. For planar motion of a monocular camera, the original method was published by Armstrong et. al. in [8]. In a more recent work [61], Gurdjos and Sturm consider the problem of camera self-calibration from images of a planar object with unknown Euclidean structure.

2.2 Video Analysis

2.2.1 Camera Motion Characterization

The recognition of parts of the video in which camera is static, zooming, or rotating, has been achieved using rather dedicated methods. Usually, these methods rely on the exploitation of motion vectors issued from block-matching techniques, or depend on the search for specific distributions of motion vectors or a few global representative motion parameters [204]. The MPEG-1 or MPEG-2 stream may also be directly exploited for camera motion characterization [136, 77], using motion vectors related to $P-$ and $B-$ frames. The method [151] is resilient to the presence of mobile objects of significative size, by computing the so-called optical flow streams built from the dominant optical flow over some extent in time. The algorithm depends, however, on many thresholds and assumes a constant camera motion type during the time extent over which optical flow streams are built. The qualitative interpretation method [13] employs divergence, curl and hyperbolic terms, expressed from a 2D affine camera motion model, for a physically meaningful interpretation of the dominant motion. Nevertheless, these 2D transformation based methods have the inherent difficulties in the cases where the scenes have dense depths and the cameras are moving, since the 2D transformation is depth dependent when the camera is not fixed. In [47], Duan et. al. develop nonparametric motion models, represented by the mean shift procedure, to overcome diverse camera shots and frequent occurrences of bad optical flow estimation.

2.2.2 Layer Segmentation

Automatic extraction of layers from a video sequence has broad applications, such as video compression and coding, recognition, and synthesis. In an earlier work, Wang and Adelson [181] propose the use of optical flow to estimate the motion layers, where each layer corresponds to a smooth motion field. Ayer and Sawhney [5] combine Minimum Description Length (MDL) and Maximum-Likelihood Estimation (MLE) in Expectation-Maximization (EM) framework to estimate the number of layers and the motion model parameters for each layer. Several other approaches [92, 175, 145] use Maximum A- Posteriori (MAP) or MLE for estimation of model parameters assuming different constraints and motion models. Another class of motion segmentation approaches group the pixels in a region by using linear subspace constraints. Ke and Kanade [87, 86] expand the seed regions into the initial layers by using k-connected components. After enforcing a low-dimensional linear affine subspace constraint on multiple frames, they cluster these initial layers into several groups and assign the image pixels to these layers. Zelnik-Manor and Irani [203] use the homography subspace for planar scenes to extract a specific layer and to register the images based on this layer.

In motion segmentation, only few researchers have tried to formulate the occlusion problem between overlapping layers. Giaccone and Jones [59] propose to use four-label system, "background, uncovered, covered and foreground", to label image pixels. The uncovered and covered pixels correspond to reappeared or occluded pixels between two frames, respectively. Based on the observation that the segmentation boundary is usually not accurate due to the occlusion, Bergen

and Meyer [15] propose to use motion estimation errors to refine the segmentation boundary but no occlusion pixel is identified in their results. In the 2D motion segmentation area, Shi and Malik [152] use the normalized graph cut to extract layers from a video sequence. Wills et al. [182] propose the use of graph cuts to extract layers between two wide baseline images. After employing the RANSAC (Random Sample Consensus) technique, they first cluster the correspondences into several initial layers, then perform the dense pixel assignment via graph cuts. Beyond the 2D motion segmentation, 3D motion (or multi-body) segmentation of multiple moving objects is another interesting topic in computer vision. In this area, the layer clustering is based on 3D geometry information, such as fundamental matrices [167] and trifocal tensors [178]. Currently, this multi-body segmentation is mainly focused on sparse point segmentation and clustering, the dense motion segmentation of 3D scene is still on the research.

2.2.3 Video Alignment

The problem of synchronizing video sequences has become an active area in computer vision community since Stein's first method [162]. Stein achieved the alignment of video sequences using tracking data obtained from multiple cameras, and assuming the cameras are static and the images are related by a 2D homography. Giese and Poggio [60] proposed a method to find the spatiotemporal alignment of two video sequences using the dynamic shift of the time stamp of the spatial information. They assumed that a 2D action trajectory can be represented as a linear-combination of prototypical views, and the effect of viewpoint changes can be expressed by varying the coef-

ficients of the linear-combination. Caspi and Irani [32] proposed a direct approach to align two surveillance videos by finding the spatiotemporal transformation that minimizes the sum of square differences between the two sequences. In [42], they also studied the problem of matching two unsynchronized video sequences of the same dynamic scene recorded by different stationary uncalibrated video cameras, based on matching space-time trajectories of moving objects. More recent methods [170, 197, 161, 34, 166, 185, 96] tackle the problem of automatic video synchronization for independently moving cameras and overlapping dynamic scenes.

The alignment of non-overlapping sequences was first addressed by Caspi and Irani [33] based on the assumption that the two sequences are captured by a stereo rig. In the two video sequences, the same motion induces the "same" changes in time. This correlated temporal behavior was used to recover spatiotemporal transformations between sequences. Wolf and Zomet [190] proposed a method for self calibrating a moving rig when the camera internal parameters are allowed to change. The relation between the optical axes of the cameras is expressed using multilinear invariants, and a solution is extracted from these invariants. Moreover, a fundamental matrix is computed for synchronizing the sequences. Rao et. al. [140] extended [42] to synchronize non-overlapping but same events, such as human activities, and were able to cope with dynamic timeline.

2.3 Video Post-production

The modification of the object-based content of a video sequence is an essential aspect of many video post-production applications [129]. This generally consists of the composition [31, 43], the removal [82, 189, 209], or the modification of the trajectories of rigid (such as a car) or nonrigid (such as a human) video objects.

2.3.1 Video Cut and Paste

In image composition, a new image, $\mathbf{I}$, can be blended from a background image, $\mathbf{B}$, and a foreground image, $\mathbf{F}$, with its alpha matte, α, by the compositing equation [129]:

$$\mathbf{I} = \alpha\mathbf{F} + (1 - \alpha)\mathbf{B}. \tag{2.1}$$

On the other hand, separation of α, $\mathbf{F}$ and $\mathbf{B}$ from a given image $\mathbf{I}$ is called matting. Compositing, as described by equation (2.1), is a straightforward operation. Matting is inherently underconstrained, since $\mathbf{F}$, $\mathbf{B}$, and $\mathbf{I}$ have three color channels each, and for each pixel we have a problem with three equations and seven unknowns. Most matting techniques solve it by controlling the background, adding images, or adding a priori assumptions about the foreground, background, and alpha. Some of these constraints and heuristics are nicely summarized by Smith and Blinn [14]. A more recent improvement on blue screen matting was developed by Mishima [112] based on representative foreground and background color samples. An alternative is to use dual-film techniques,

such as the infrared (IR) matting system designed by Debevec et al. [46] and the invisible-signal keying system developed by Ben-Ezra [11].

When filming with specialized backdrops is impossible or impractical, it is necessary to pull a matte from a photograph of the foreground object taken with a natural background. This problem is called *natural image matting*. Rotoscoping is a commonly used technique for solving this problem. Mitsunaga et al. [120] developed the AutoKey system to improve the rotoscoping process for video matting. Agarwala et al. proposed a keyframe-based system to effectively reduce the amount of human effort for rotoscoping [6]. Recently, there are also further developments in natural image matting and compositing. The statistical approaches [143, 24, 72, 22, 3] are proposed to to address this deficiency of the rotoscoping. Sun et. al. [150] formulated the problem of natural image matting as one of solving Poisson equations with the matte gradient field, while Rother et. al. solved this problem using the graph-cut based optimization [141]. Zongker et al. introduced environment matting and compositing to generalize the traditional matting and compositing processes to incorporate refraction and reflection [208]. Following their work, Chuang et. al. proposed a novel compositing model for shadows and a practical process for shadow matting and compositing [30], and thus improved the environment matting algorithm to acquire more accurate environment mattes.

2.3.2 Video Completion

Another important task in video post-production is video completion. The goal is to repair the missing pixels in the holes created by damage to the video or removal of selected objects. Typically, this goal is achieved by prediction from information in the undamaged frames. Bertalmo et. al. [10] proposed a frame-by-frame PDEs based video inpainting approach, which extends image inpainting techniques to video sequences. Wexler et. al. [189] described a method for space-time completion of large space-time "holes" in video sequences, in which they treat video completion as a global optimization problem, and enforce global spatio-temporal coherence by a well-defined objective function. A similar work was proposed by in et. al. [76].

Patwardhan et. al. [137] extended the image inpainting techniques proposed in [35] to video inpainting by assigning priority to spatio-temporal locations in the video, and copy the spatial patch with highest priority to holes in the background/foreground frame by frame. Jia et. al. [82] proposed an approach to repair videos with periodically moving object under static/moving camera. In their framework, the background is separated into multiple layers, which are repaired individually by a layered mosaics approach and then merge together. A homography blending technique is also used to remove small holes and overlapping of the boundaries of different layers. When repairing foreground of periodic motion, sample moving elements or movels, which describe the periodic characteristics of motion, are extracted. The missing motion pixels are then repaired by aligning sample movels to the damaged movel, which also ensures temporal coherence in the resulting video.

CHAPTER 3

BACKGROUND GEOMETRY

3.1 Introduction

Projective geometry, the theory of invariants of the group of projective transformations [157], provides this thesis with the basic mathematical background upon which an effective and robust video analysis and post-production framework is developed. If Euclidean geometry is interpreted as the geometry of the straight edge and compass, projective geometry is the geometry of the straight edge alone. In a nutshell projective geometry is linear algebra disguised as geometry.

This chapter presents a brief review of some backgrounds of projective geometry that will be necessary for understanding the remainder of this dissertation. No attempt has been made to provide a comprehensive survey, which can be found in many good references on projective geometry [157] and computer vision [49, 55, 75]. Although the treatment of the subject presented herein is non-standard and rather advanced topics are covered without preliminary introduction, a reader proficient in projective geometry may want to skip this chapter and proceed to Chapter 4.

3.2 History

The drop from three-dimensional world to a two-dimensional image is a projection process in which we lose one dimension. The usual way of modelling this process is by central or perspective projection in which a ray from a point in space is drawn from a 3D world point through a fixed point in space, called the center of projection. Perspective projection is a framework that is used by artists, designers, engineers, etc. to represent three-dimensional objects on a two-dimensional surface. An artist uses perspective projection to represent nature or objects in the most effective way possible. It evolved from "Construzione Legittma" that was probably invented in the early fifteenth century, most likely by Fillipo Brunelleschi. Leon Battista Alberti, Uccello and Piero della Francesca all improved upon Brunelleschi's theories.

These pre-renaissance and renaissance artists have noticed that although parallel lines never meet in a Euclidean space, their perspective projections may do at a so-called vanishing point on the horizon. Such early observations about perspective projection were based on a single vanishing point, and any other parallel lines were exempted from the idea that they had to meet at some point in the distance.

Prior to Wollaston's Camera Lucida of 1806 and Varley's Graphic Telescope of 1811, there was a range of devices developed around the 15th and 16th centuries to aid the artists in correctly illustrating linear perspective. Leon Battista Alberti, Leonardo Da Vinci and later Albrecht Drer and the lesser known Jacob de Keyser developed these devices. One example is shown in Figure 3.1.

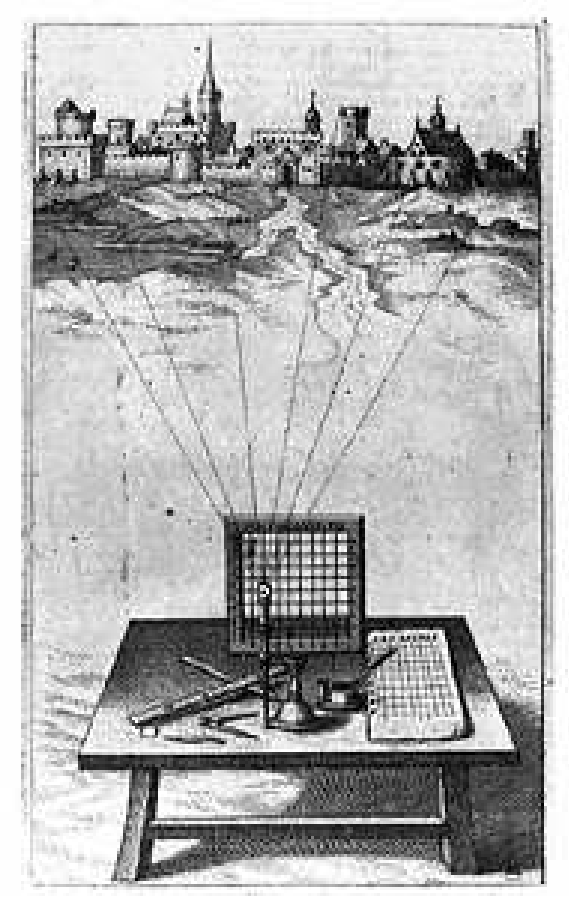


Figure 3.1: Alberti's Grid (also called Alberti's Veil), 1450, also known as The Square Grid of the Renaissance.

3.3 Notations

This thesis employs quite a standard notation convention, mostly consistent to the textbook by Hartley and Zisserman [75]:

- 3D points and lines in general position are denoted by upper case bold symbols (e.g. $\mathbf{X}$, $\mathbf{L}$);
- 3D (2D) planes are denoted by upper (lower) case bold Greek symbols, e.g. $\mathbf{\Pi}$ ($\boldsymbol{\pi}$);

- image positions and vectors by lower case bold symbols (e.g. $\mathbf{x}$, $\mathbf{l}$);
- values and scalars by normal face symbols (e.g. f, λ and s);
- matrices by upper case bold symbols (e.g. $\mathbf{H}$, $\mathbf{P}$);
- sets by symbols in "script" or "caligraph" font (e.g. $\mathcal{S}$);
- The vectors are denoted in homogeneous coordinate system[1].
- equality up to multiplication by a non-zero scale factor is indicated by $\sim$. This equality is very typical in homogeneous system.

When necessary, further notation choices are described in each chapter.

3.4 Pin-hole Camera Model

A real world camera can be modeled by a pin-hole or perspective camera model. More comprehensive imaging models including single viewpoint [69] and non-single viewpoint [147] are out of the scope of this thesis. Algebraically, a pin-hole camera (see Figure 3.2) projects a region of $\mathbb{R}^3$ lying in front of the camera into a region of the image plane $\mathbb{R}^2$, based on the principle of collinearity. As

[1]Consider a point $\mathbf{X}$ in the n-dimensional space with Cartesian coordinates given by the n-tuple $(X_1, X_2, \cdots, X_n) \in \mathbb{R}^n$, the expression of $\mathbf{X}$ in homogeneous coordinates is the set of $(n+1)$-tuple $\{w(X_1, X_2, \cdots, X_n, 1), \forall w \in \mathbb{R}\backslash\{0\}\}$. Conversely, given the homogeneous coordinates $\{w(X_1, X_2, \cdots, X_n, X_{n+1}), \forall w \in \mathbb{R}\backslash\{0\}\}$ of a point $\mathbf{X}$ in the n-dimensional space, the Cartesian coordinates of $\mathbf{X}$ will be given by $(X_1, X_2, \cdots, X_n)/X_{n+1}$, if $X_{n+1} \neq 0$. If $X_{n+1} = 0$, the point $\mathbf{X}$ is said to be at infinity in direction $(X_1, X_2, \cdots, X_n)$, and it cannot be represented in Cartesian coordinates

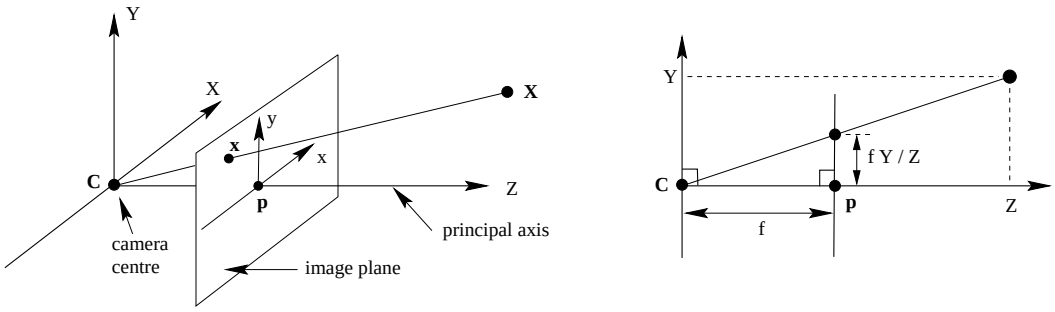


Figure 3.2: **Pinhole camera geometry.** $\mathbf{C}$ is the camera center and $\mathbf{p}$ the principal point. The camera center is here placed at the coordinate origin. Note that the image plane is placed in front of the camera center. Figures courtesy of Hartley and Zisserman [64].

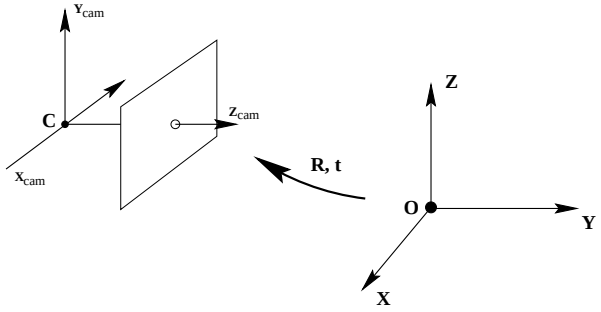


Figure 3.3: The Euclidean transformation between the world and camera coordinate frames. Figures courtesy of Hartley and Zisserman [64].

is well known, a 3D point $\mathbf{M} = [X\ Y\ Z\ 1]^T$ and its corresponding image projection $\mathbf{m} = [u\ v\ 1]^T$ are related via a 3×4 matrix $\mathbf{P}$ as

$$\mathbf{m} \sim \underbrace{\mathbf{K}[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3\ \mathbf{t}]}_{\mathbf{P}}\mathbf{M}, \quad \mathbf{K} = \begin{bmatrix} f & \gamma & u_0 \\ 0 & \lambda f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.1}$$

where $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ are the columns of the 3×3 orthonormal rotation matrix $\mathbf{R}$, $\mathbf{t} = -\mathbf{R}\mathbf{C}$, with $\mathbf{C} = [C_x\ C_y\ C_z]^T$ representing the coordinates of the camera center in the world coordinate frame,

is the translation vector, and $\mathbf{K}$ is a nonsingular 3×3 upper triangular matrix known as the camera calibration matrix including five parameters: the focal length f, the skew γ, the aspect ratio λ and the principal point at $(u_0,\ v_0)$.

If the image axes are orthogonal to each other, which is often the case [75], γ will be equal to zero. The intrinsic parameters in $\mathbf{K}$ define the internal imaging geometry of the camera, while the extrinsic parameters ($\mathbf{R}$ and $\mathbf{t}$, see Figure 3.3) relate the world coordinate frame to that of the camera. Camera calibration is the process of estimating these parameters. A camera is said to be calibrated if its intrinsic parameters are known. If both the intrinsic and the extrinsic parameters of a camera are known, then the camera is said to be fully calibrated.

3.5 Perspective Mapping

3.5.1 Planar Homography

An interesting specialization of the general central projection described above is a plane-to-plane projection or a 2D-2D projective mapping. Points on a plane are mapped to points on another plane by a plane-to-plane homography, also known as a planar projective transformation. It is a bijective (thus invertible) mapping induced by the star of rays centered in the camera center (center of projection). Planar homographies arise, for instance, when a world planar surface is imaged. A planar surface viewed from two different viewpoints induces a homography between the two

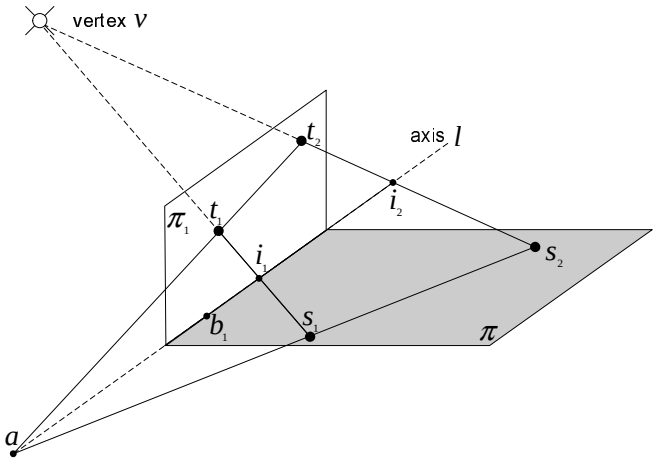


Figure 3.4: Geometrically, a planar object, π_1, and its shadow, illuminated by a point light source $\mathbf{v}$ and cast on a ground plane π, are related by a planar homology.

images. Points on the world plane can be transferred from one image to the other by means of a homography mapping.

3.5.2 Planar Homology

A *planar homology* is a special planar projective transformation which has a line of fixed points, called the *axis*, and a distinct fixed point $\mathbf{v}$, not on the axis $\mathbf{l}$, called the *vertex* of the homology. In the error-free case, imaged shadow relations (illuminated by a point light source) are modeled by a *planar homology* [157, 180, 41] (see Figure 3.4). In this imaged shadow relation, the vertex $\mathbf{v}$ is the image of the light source, and the axis, $\mathbf{l}$, is the image of the intersection of the vertical plane π_1 and the ground plane π, on which π_1 casts its shadow. Under this transformation, points on the axis are mapped to themselves. Each point off the axis, e.g. $\mathbf{t}_2$, lies on a fixed line $\mathbf{t}_2\mathbf{s}_2$

through $\mathbf{v}$ intersecting the axis at $\mathbf{i}_2$ and is mapped to another point $\mathbf{s}_2$ on the line. Note that $\mathbf{i}_2$ is the intersection in the image plane, although the light ray $\mathbf{t}_2\mathbf{s}_2$ and the axis, $\mathbf{l}$, are unlikely to intersect in the 3D world. Note also that the light source $\mathbf{v}$ does not have to be infinite to keep the model as a planar homology, as long as the light source is a point light source, i.e. all light rays are concurrent.

One important property of a planar homology is that the corresponding lines, i.e. lines through pairs of corresponding points, intersect with the axis: for example, the lines $\mathbf{t}_1\mathbf{t}_2$ and $\mathbf{s}_1\mathbf{s}_2$ intersect at a point $\mathbf{a}$ on the axis $\mathbf{l}$. Another important property of a planar homology is that the cross ratio defined by the vertex, $\mathbf{v}$, the corresponding points, $\mathbf{t}_i$ and $\mathbf{s}_i$, and the intersection, $\mathbf{i}_i$, of the line $\mathbf{t}_i\mathbf{s}_i$ with the axis, is the characteristic invariance of the homology, and is the same for all corresponding points. For example, the cross ratios of the four points $\{\mathbf{v}, \mathbf{t}_1; \mathbf{s}_1, \mathbf{i}_1\}$ and $\{\mathbf{v}, \mathbf{t}_2; \mathbf{s}_2, \mathbf{i}_2\}$ are equal.

3.6 Image of the Absolute Conic

Before the introduction of the image of the absolute conic, the expression of a conic in matrix is first described. Consider the equation

$$ax_1^2 + 2bx_1x_2 + 2cx_1 + dx_2^2 + 2ex_2 + f = 0 \tag{3.2}$$

of a point conic C. In homogeneous coordinates, (3.2) becomes

$$\mathbf{x}^T\mathbf{C}\mathbf{x} = 0, \tag{3.3}$$

where $\mathbf{x} = [x_1\ x_2\ 1]^T$ and

$$\mathbf{C} \sim \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix}. \tag{3.4}$$

The matrix $\mathbf{C}$ is the representation of the conic C in homogeneous coordinates. Note that the matrix $\mathbf{C}$ is always symmetric.

The *Image of the Absolute Conic* $\boldsymbol{\omega}$ is an imaginary point conic directly related to the camera internal matrix $\mathbf{K}$ in Equation (3.1) by $\boldsymbol{\omega} = \mathbf{K}^{-T}\mathbf{K}^{-1}$, which can be expanded up to a non-zero scale, f^2, as:

$$\boldsymbol{\omega} \sim \begin{bmatrix} 1 & -\frac{\gamma}{f\lambda} & \frac{\gamma v_0 - \lambda f u_0}{f\lambda} \\ -\frac{\gamma}{f\lambda} & \frac{f^2+\gamma^2}{f^2\lambda^2} & -\frac{\gamma^2 v_0 - \gamma\lambda f u_0 + v_0 f^2}{f^2\lambda^2} \\ \frac{\gamma v_0 - \lambda f u_0}{f\lambda} & -\frac{\gamma^2 v_0 - \gamma\lambda f u_0 + v_0 f^2}{f^2\lambda^2} & \frac{v_0^2(f^2+\gamma^2) - 2\gamma v_0 \lambda f u_0}{f^2\lambda^2} + f^2 + u_0^2 \end{bmatrix}. \tag{3.5}$$

Instead of directly determining $\mathbf{K}$, it is possible to compute the symmetric matrix $\boldsymbol{\omega}$ or its inverse (the dual image of the absolute conic), and then compute the calibration matrix using either Cholesky factorization [131] or uniquely as [200, 40]

$$\begin{aligned} \lambda &= \sqrt{1/(\omega_{22} - \omega_{12}^2)}\ , \\ v_0 &= (\omega_{12}\omega_{13} - \omega_{23})/(\omega_{22} - \omega_{12}^2), \\ u_0 &= -(v_0\omega_{12} + \omega_{13}), \\ f &= \sqrt{\omega_{33} - \omega_{13}^2 - v_0(\omega_{12}\omega_{13} - \omega_{23})}\ , \\ \gamma &= -f\lambda\omega_{12}, \end{aligned} \tag{3.6}$$

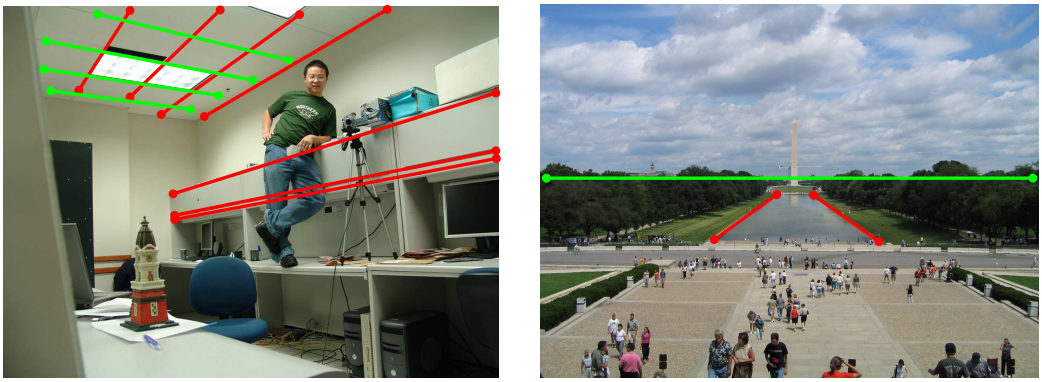


Figure 3.5: Examples of vanishing points in real images and paintings. The vanishing points can be computed by intersecting image lines along parallel directions, which are plotted in the same color. The green line in the right is the horizon line.

where the subscripts of ω_{ij} denote the element's row (i) and column (j) in the matrix $\boldsymbol{\omega}$. This typically leads to simple, and in particular, linear calibration equations as shown next.

3.7 Calibration from Vanishing Points

Parallel world lines, such as railway lines, are imaged as converging lines, and their intersection is the *vanishing point* for the direction of the railway. Similarly, parallel world planes intersect at a *vanishing line*, on which the vanishing points lie. Horizon line is a special vanishing line, the intersection of ground plane and sky (see Figure 3.5). Vanishing points and vanishing lines are extremely powerful geometric cues. They convey a wealth of information about direction of lines and orientation of planes. These entities can be estimated directly from the images and no explicit knowledge of the relative geometry between camera and viewed scene is required [113, 104, 149,

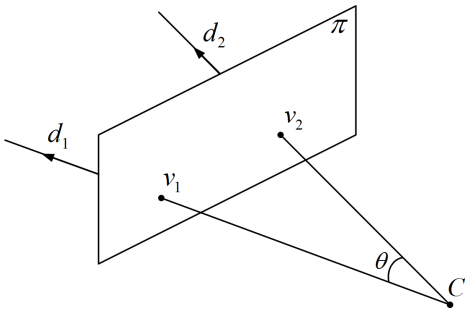


Figure 3.6: The vanishing points of the world lines with direction $\mathbf{d}_1$ and $\mathbf{d}_2$ in 3-space are the intersections $\mathbf{v}_1$ and $\mathbf{v}_2$ of the image plane $\boldsymbol{\pi}$ with the rays through the camera $\mathbf{C}$ with directions $\mathbf{d}_1$ and $\mathbf{d}_2$. θ is the angle between two rays $\mathbf{d}_1$ and $\mathbf{d}_2$.

179]. Sometimes, they lie outside the physical boundaries of images or paintings. But this does not affect the computations.

In [105], Liebowitz and Zisserman formulated the calibration constraints provided by vanishing points of mutually orthogonal directions in terms of the geometry of the $\boldsymbol{\omega}$. These intra-image orthogonal constraints, together with our novel inter-image constraint associated with the "*weak*" pole-polar relationship, will be used later in Section 4.2 to derive a simple technique for camera calibration from shadows. The weak pole-polar relationship here means that the two independent constraints on $\boldsymbol{\omega}$ arising from the pole-polar relationship can not be identified from a single view. A simple derivation of Liebowitz and Zisserman's result is given below.

In projective 3-space, the plane at infinity, $\boldsymbol{\pi}_\infty$, is the plane of directions, and all lines with the same direction intersect $\boldsymbol{\pi}_\infty$ in the same point. The vanishing point is simply the image of this intersection (see Figure 3.6). Therefore, if a line has direction $\mathbf{d}$, then it intersects $\boldsymbol{\pi}_\infty$ at the point $\mathbf{X}_\infty = [\mathbf{d}^T \ 0]^T$. Consequently, the vanishing point, $\mathbf{v}$, of the lines with direction $\mathbf{d}$ is the image

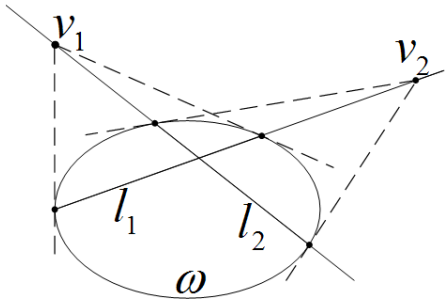


Figure 3.7: Points $\mathbf{v}_1$ and $\mathbf{v}_2$, with polars $\mathbf{l}_1$ and $\mathbf{l}_2$ respectively. Since point $\mathbf{v}_1$ lies on the polar of point $\mathbf{v}_2$, the points $\mathbf{v}_1$ and $\mathbf{v}_2$ are conjugate with respect to the conic $\boldsymbol{\omega}$, and vice versa.

of $\mathbf{X}_\infty$ (for simplicity, the world coordinate frame will be chosen to coincide with the camera coordinate frame) [75]

$$\mathbf{v} \sim \mathbf{P}\mathbf{X}_\infty = \mathbf{K}[\mathbf{I} \mid \mathbf{0}][\mathbf{d}^T \; 0]^T = \mathbf{K}\mathbf{d}. \tag{3.7}$$

Conversely, the direction $\mathbf{d}$ is obtained from the vanishing point $\mathbf{v}$ as $\mathbf{d} = \mathbf{K}^{-1}\mathbf{v}$ up to a scale. Note that $\mathbf{v}$ depends only on the direction $\mathbf{d}$ of the line, not on its position. The angle between two rays, with directions $\mathbf{d}_1$ and $\mathbf{d}_2$ corresponding to vanishing points $\mathbf{v}_1$ and $\mathbf{v}_2$ respectively, may be obtained from the cosine formula for the angle between two vectors:

$$\begin{aligned} \cos(\theta) &= \frac{\mathbf{d}_1^T\mathbf{d}_2}{\sqrt{\mathbf{d}_1^T\mathbf{d}_1}\sqrt{\mathbf{d}_2^T\mathbf{d}_2}} \\ &= \frac{(\mathbf{K}^{-1}\mathbf{v}_1)^T(\mathbf{K}^{-1}\mathbf{v}_2)}{\sqrt{(\mathbf{K}^{-1}\mathbf{v}_1)^T(\mathbf{K}^{-1}\mathbf{v}_1)}\sqrt{(\mathbf{K}^{-1}\mathbf{v}_2)^T(\mathbf{K}^{-1}\mathbf{v}_2)}} \\ &= \frac{\mathbf{v}_1^T\boldsymbol{\omega}\mathbf{v}_2}{\sqrt{\mathbf{v}_1^T\boldsymbol{\omega}\mathbf{v}_1}\sqrt{\mathbf{v}_2^T\boldsymbol{\omega}\mathbf{v}_2}}. \end{aligned} \tag{3.8}$$

In practice, it is often the case that the lines and planes which give rise to vanishing points are orthogonal, i.e. $\cos(\theta)$ in Equation (3.8) is zero. In this case the vanishing points, $\mathbf{v}_1$ and $\mathbf{v}_2$, of two perpendicular world lines satisfy $\mathbf{v}_1^T\boldsymbol{\omega}\mathbf{v}_2 = 0$. Geometrically, the two points $\mathbf{v}_1$ and $\mathbf{v}_2$ are

said to be conjugate with respect to the conic $\boldsymbol{\omega}$, which is shown in Figure 3.7. The orthogonality relations provide equations that are linear in the elements of $\boldsymbol{\omega}$, and hence can be used to calibrate the camera, e.g. [25, 105, 200, 153, 186, 23]. The earlier result reported by Caprile and Torre in [44] is a special case where the camera has zero skew and unit aspect ratio and, thus, $\boldsymbol{\omega}$ in Equation (3.5) degenerates to

$$\boldsymbol{\omega} \sim \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ -u_0 & -v_0 & u_0^2 + v_0^2 + f^2 \end{bmatrix}. \tag{3.9}$$

Equivalently, $\boldsymbol{\omega}$ is the conic

$$(x - u_0)^2 + (y - v_0)^2 + f^2 = 0, \tag{3.10}$$

which may be interpreted as a circle aligned with the axes, centered on the principal point, and with radius if.

Zhang's flexible calibration method [200] uses the orthogonal property of the columns $\mathbf{r}_1$ and $\mathbf{r}_2$ of the rotation matrix $\mathbf{R}$ in Equation (3.1), i.e. $\mathbf{p}_1^T \boldsymbol{\omega} \mathbf{p}_2 = 0$, where $\mathbf{p}_i$ ($\mathbf{h}_i$ in Zhang's notation [200], $i = 1, 2$) denotes the i^{th} column of the projection matrix $\mathbf{P}$ in Equation (3.1). This is actually equivalent to the constraint $\mathbf{v}_x^T \boldsymbol{\omega} \mathbf{v}_y = 0$, where $\mathbf{v}_x$ (respectively $\mathbf{v}_y$) is the vanishing point along the world X-axis (respectively Y-axis) direction. To better understand this, note that $\mathbf{p}_1$ is the scaled vanishing point $\mathbf{v}_x$ of the ray along the world X-axis direction $[1 \;\; 0 \;\; 0]^T$,

$$\mathbf{v}_x \sim [\mathbf{p}_1 \;\; \mathbf{p}_2 \;\; \mathbf{p}_3 \;\; \mathbf{p}_4][1 \;\; 0 \;\; 0 \;\; 0]^T = \mathbf{p}_1. \tag{3.11}$$

Similarly, $\mathbf{p}_2$ is scaled vanishing point $\mathbf{v}_y$ of the ray along the world Y-axis direction $[0 \;\; 1 \;\; 0]^T$. It is important to mention that the above derivation does not apply to the normal property of $\mathbf{r}_1$

and $\mathbf{r}_2$, i.e. $\mathbf{p}_1^T\boldsymbol{\omega}\mathbf{p}_1 - \mathbf{p}_2^T\boldsymbol{\omega}\mathbf{p}_2 = 0$ used in [200, 153], since there are likely different scales for $\mathbf{v}_x$ and $\mathbf{v}_y$. They [200, 153] are able to determine these scales since they know the world to image homographies. In [186, 23], they utilize the pole-polar relationship with respect to $\boldsymbol{\omega}$: the vanishing point $\mathbf{v}$ of the normal direction to a plane is related to the plane vanishing line $\mathbf{l}$ as $\mathbf{l} = \boldsymbol{\omega}\mathbf{v}$. Since any vanishing point $\mathbf{x}$ that is orthogonal to $\mathbf{v}$ must satisfy $\mathbf{x}^T\boldsymbol{\omega}\mathbf{v} = 0$, the important point now is that this is true for any point $\mathbf{x}$ satisfying $\mathbf{x}^T\mathbf{l} = 0$, and it follows that $\mathbf{l} = \boldsymbol{\omega}\mathbf{v}$.

3.8 Epipolar geometry

The previous subsections have discussed the single view geometry. Among the geometric properties of a set of two cameras, the widely known property in computer vision is the epipolar geometry [49]. It is algebraically represented by the fundamental matrix $\mathbf{F}$,

$$\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0, \tag{3.12}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are a pair of corresponding points in two images. $\mathbf{F}$ is also known as the uncalibrated version of the essential matrix [49], $\mathbf{E}$, because

$$\mathbf{F} = \mathbf{K}'^{-T}\mathbf{E}\mathbf{K}^{-1}, \tag{3.13}$$

where $\mathbf{K}'$ and $\mathbf{K}$ are matrices representing the internal calibration parameters of the stereo cameras. In general, both matrices $\mathbf{F}$ and $\mathbf{E}$ are of rank two. For an arbitrary stereo pair, the rank two constraint is the only constraint on $\mathbf{F}$, and thus $\mathbf{F}$ generally has seven degrees of freedom. The essential matrix, $\mathbf{E} = [\mathbf{t}]_\times\mathbf{R}$, where the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ represent

the motion between the two positions and orientations of the cameras, has only five degrees of freedom, since both $\mathbf{R}$ and $\mathbf{t}$ have three degrees of freedom, but there is an overall scale ambiguity. Note that the essential matrix, $\mathbf{E}$, must have a zero singular value and two equal nonzero singular values, which is also known as the Huang-Faugeras constraint [70, 65].

CHAPTER 4

CAMERA CALIBRATION USING OBJECTS

This chapter proposes new inter-image calibration constraints, which extend the current state-of-the-art calibration techniques to situations where only one vanishing point is known. Therefore, these constraints are particularly useful for scenes where only minimal information is available. First, two vertical objects and their parallel cast shadows on the planar ground plane are used as a configuration to demonstrate the efficiency of the inter-image constraints. In this configuration, the camera parameters and the orientation of an infinite light source, e.g. the sun, can be recovered. The implementation details and degenerate configurations of the proposed algorithm are also described. Second, the inter-image constraints are applied to symmetric objects and 1D objects for camera calibration. Finally, applications to reconstruction of partially viewed symmetric objects and image based metrology are demonstrated.

4.1 Introduction

The camera calibration method using vertical objects and their cast shadows introduced in this chapter relates to the existing methods [44, 200, 105, 153, 106, 186, 23, 89] that explore the calibration constraints provided by vanishing points of orthogonal directions or pole-polar relationship in terms of the geometry of the Image of the Absolute Conic (see Section 3.7). Similar to other calibration methods from vanishing points, which require only the presence of mutually orthogonal directions, the technique presented herein requires only the calibration target to be vertical objects and their parallel cast shadows on the ground plane, which provide two mutually orthogonal directions. However, we relaxe some of the constraints imposed in other calibration algorithms such as [105], which requires minimum four views of the above configuration to determine the aspect ratio, the focal length, and the principal point of the camera. To this end we propose novel inter-image constraints on the camera intrinsic parameters. Generally, it is shown that two corresponding image projections of the same 3D line on the ground plane provides two constraints on the camera intrinsic parameters. Consequently, two views are sufficient to determine the aspect ratio, the focal length, and the principal point.

The proposed techniques fall halfway between calibration from known structures and full fledged self-calibration [186], i.e. they require calibration objects, but do not need the measurements of any distance or angle in the 3D world. Although some recent efforts using identifiable targets of known shape such as architectural buildings [105], surfaces of revolution [186, 23] and circles [45] are toward the similar goal, we argue that the alternative calibration objects, such as

symmetric objects, and vertical objects and their cast shadows, are also common in the real world. Note that effects such as radial distortion (often arising in slightly wide-angle lenses typically used in the security cameras) which corrupt the central projection model can generally be removed [75], and are therefore not detrimental to our methods.

The remainder of this chapter is organized as follows. We describe in section 4.2 the details of the method based on using vertical objects and their cast shadows, followed by the implementation details in Section 4.3 and the discussion of the singular configurations in Section 4.4. In Section 4.5, we demonstrate the results of the method on both synthesized and real images. In addition, we show examples of the inter-image constraints applied to other calibration configurations including symmetric and 1D objects. Applications to reconstruction of partially viewed symmetric objects and image-based metrology are shown in Section 4.6. Finally, Section 4.7 concludes this chapter with observations and proposed areas of future work.

4.2 Our Method

In this section, vertical objects and their parallel cast shadows on the planar ground plane are used as a calibration configuration to demonstrate the efficiency of the inter-image constraints.

Shadows are important because they provide important visual cues, e.g. depth and shape, in our perception of the world [132]. They are also useful for computer vision applications such as building detection [98], surveillance system [154, 71, 108, 80, 135, 121] and inverse lighting

[160, 127]. However, shadows are not frequently addressed in 3D vision. Some few examples include the work on the relationship between the shadows and object structure by Kriegman and Belhumeur [83], the work on object recognition by Van Gool et. al. [180], and the work on weak structured lighting by Bouguet and Perona [18]. Shadows are interesting for camera calibration since the shadows of two parallel vertical lines cast on the planar ground plane by an infinite point light source are also parallel, which together with the vertical vanishing point provide orthogonal constraint on the image of the absolute conic.

Our geometric constructions are very close to the ones in [142] and [1]. However, our work is different from Reid and North's work [142] in three aspects: first, we aim at estimating the camera parameters and light source orientation, while they focus on the problem of affine reconstruction of a ball out of the ground plane. Second, they [142] have the input of the image of the ground plane's horizon line, and we have, rather than the horizon line, the bottom point of the second vertical object. Third, our major contribution is the inter-image constraints on the IAC, and we use two images. Although Antone and Bosse [1] made use of shadows for camera calibration and have the similar setup as ours, they mainly utilize the world-to-camera correspondences, and involve knowledge of absolute 3D quantities. Different from their techniques, the contribution of our method lies on the relief of the requirement of knowledge of 3D quantities.

This section first examines the geometry of the scenes and the orthogonal constrains available from each view. Then we introduce the novel inter-image constraints associated with the weak pole-polar relationships. Finally, the orientation of the light source can also be computed.

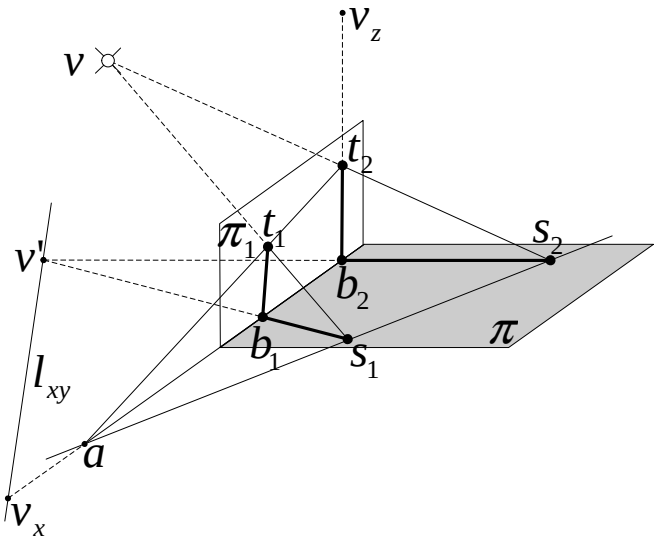


Figure 4.1: Basic geometry of a scene with two vertical lines $\mathbf{t}_1\mathbf{b}_1$ and $\mathbf{t}_2\mathbf{b}_2$ casting shadows $\mathbf{s}_1\mathbf{b}_1$ and $\mathbf{s}_2\mathbf{b}_2$ on the ground plane $\boldsymbol{\pi}$ by the distant light source $\mathbf{v}$. $\boldsymbol{\pi}_1$ is the vertical plane consisting of two parallel lines $\mathbf{t}_1\mathbf{b}_1$ and $\mathbf{t}_2\mathbf{b}_2$. $\mathbf{l}_{xy}$ is the vanishing line of the ground plane $\boldsymbol{\pi}$, and thus passes through the intersection of the two cast shadow lines $\mathbf{s}_1\mathbf{b}_1$ and $\mathbf{s}_2\mathbf{b}_2$.

4.2.1 Scene Geometry and Orthogonal Constraints

The basic geometry of a scene containing two vertical lines and their shadows on the ground plane cast by an infinite point light source is shown in Figure 4.1. Note that this figure shows the projections of the 3D world points in the image plane denoted by corresponding lower-case characters. For example, the world point $\mathbf{B}_2$ (not shown in Figure 4.1) is mapped to $\mathbf{b}_2$ in the image plane. Below, we first explore the constraints available from a single view, given the above configuration.

First, we can compute the vanishing point $\mathbf{v}_z$ along the vertical direction by intersecting the two vertical lines as

$$\mathbf{v}_z \sim (\mathbf{t}_1 \times \mathbf{b}_1) \times (\mathbf{t}_2 \times \mathbf{b}_2). \tag{4.1}$$

Since the light source, $\mathbf{v}$, is at infinity, the two shadow lines $\mathbf{S}_2\mathbf{B}_2$ and $\mathbf{S}_1\mathbf{B}_1$ must be parallel in the 3D world. In other words, the two imaged parallel shadow lines will intersect in the image space at the vanishing point $\mathbf{v}'$

$$\mathbf{v}' \sim (\mathbf{s}_1 \times \mathbf{b}_1) \times (\mathbf{s}_2 \times \mathbf{b}_2). \tag{4.2}$$

From the pole-polar relationship with respect to the Image of the Absolute Conic $\boldsymbol{\omega}$ (see Section 3.7), the vanishing point $\mathbf{v}_z$ of the normal direction to a plane (ground plane $\boldsymbol{\pi}$ in our case) is the pole to the polar which is the vanishing line $\mathbf{l}_{xy}$ of the plane,

$$\mathbf{v}_x \times \mathbf{v}' \sim \mathbf{l}_{yz} \sim \boldsymbol{\omega}\mathbf{v}_z, \tag{4.3}$$

where $\mathbf{v}_x$ is the intersection of the line $\mathbf{b}_1\mathbf{b}_2$ and the vanishing line $\mathbf{l}_{xy}$, i.e. another vanishing point on the ground plane. Equation (4.3) can be rewritten, equivalently, as two constraints on the Image of the Absolute Conic $\boldsymbol{\omega}$:

$$\mathbf{v}'^T\boldsymbol{\omega}\mathbf{v}_z = 0, \tag{4.4}$$

$$\mathbf{v}_x^T\boldsymbol{\omega}\mathbf{v}_z = 0. \tag{4.5}$$

In our case, we only have the constraint (4.4) since we can not determine $\mathbf{v}_x$ from a single view. Without further assumptions, we are unlikely to extract more constraints on the camera internal parameters from a single view of such a scene as shown in Figure 4.1.

Before we move further, we mention some possible configurations that may provide more constraints, although we will not make use of such constraints. One possibility is to use the ratio between the heights h_1 and h_2 of the two points $\mathbf{t}_1$ and $\mathbf{t}_2$ from the ground plane since

$$\frac{h_1}{h_2} = \frac{(\mathbf{a} - \mathbf{b}_1)(\mathbf{v}_x - \mathbf{b}_2)}{(\mathbf{a} - \mathbf{b}_2)(\mathbf{v}_x - \mathbf{b}_1)}, \tag{4.6}$$

where $\mathbf{a} \sim (\mathbf{t}_1 \times \mathbf{t}_2) \times (\mathbf{s}_1 \times \mathbf{s}_2)$ lies on the line $\mathbf{b}_1\mathbf{b}_2$. Equation (4.6) can be directly used to recover the second vanishing point $\mathbf{v}_x$ on the horizon line $\mathbf{l}_{xy}$. One special case is that $\mathbf{t}_1$ and $\mathbf{t}_1$ have the same height from the ground plane, in which case the vanishing point $\mathbf{v}_1$ coincide with the intersection $\mathbf{a}$. Other possibilities include utilizing the knowledge of the orientation of the light source $\mathbf{v}$ as shown in [1]. These knowledge provides more constraints than the one we have in Equation (4.4), and therefore it is possible to calibrate the camera using one view by assuming a simplified camera model [44, 105]. However, too many assumptions limit the applicabilities in the real world. To alleviate this problem we propose to use an additional view and the resulting inter-image constraints.

4.2.2 Inter-image Constraints

The second view can be easily used to obtain one more constraint on $\boldsymbol{\omega}$ from Equation (4.4). Beyond that, we explore here more constraints based on the inter-image constraints associated with the weak pole-polar relationship.

Geometrically, equation (4.5) can be interpreted as the constraint that the vanishing point $\mathbf{v}_x$ of the direction $\mathbf{B}_1\mathbf{B}_2$ must lie on the line $\boldsymbol{\omega}\mathbf{v}_z$, which is the vanishing line $\mathbf{l}_{xy}$ of the ground plane. Considering also that $\mathbf{v}_x$ lies on the imaged line $\mathbf{b}_1\mathbf{b}_2$, we can express $\mathbf{v}_x$ as a function of $\boldsymbol{\omega}$:

$$\mathbf{v}_x \sim [\mathbf{b}_1 \times \mathbf{b}_2]_\times \boldsymbol{\omega}\mathbf{v}_z, \tag{4.7}$$

where $[\cdot]_\times$ is the notation for the skew symmetric matrix [49] characterizing the cross product. We call this relationship as the weak pole-polar relationship in the sense that we can not identify the polar, $\mathbf{l}_{xy} = \boldsymbol{\omega}\mathbf{v}_z$, to the pole $\mathbf{v}_z$ with respect to $\boldsymbol{\omega}$ from a single view, since we only have one constraint $\mathbf{v}'^T\mathbf{l}_{xy} = 0$ on $\mathbf{l}_{xy}$.

However, there are inter-image constraints associated with the weak pole-polar relationship (4.7) as shown below. We have two planes $\boldsymbol{\pi}$ and $\boldsymbol{\pi}_1$ as shown in Figure 4.1. The inter-image planar homography $\mathbf{H}_{\boldsymbol{\pi}}$ corresponding to $\boldsymbol{\pi}$ can be computed from at least four pairs of image points $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{b}_1$ and $\mathbf{b}_2$, while $\mathbf{H}_{\boldsymbol{\pi}_1}$ corresponding to $\boldsymbol{\pi}_1$ can be computed from $\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{b}_1$ and $\mathbf{b}_2$. In addition, the Fundamental matrix $\mathbf{F}$ between the two views can be computed using the methods [138, 199] that minimize the reprojection errors, provided that there are more point correspondences. Therefore, we have the following inter-image constraints in terms of $\boldsymbol{\omega}$,

$$\mathbf{v}'_x \sim \mathbf{H}_{\boldsymbol{\pi}}\mathbf{v}_x, \tag{4.8}$$

$$\mathbf{v}'_x \sim \mathbf{H}_{\boldsymbol{\pi}_1}\mathbf{v}_x \tag{4.9}$$

$$\mathbf{v}'^T_x\mathbf{F}\mathbf{v}_x = 0, \tag{4.10}$$

where $\mathbf{v}'_x$ is the corresponding vanishing point of $\mathbf{v}_x$ in the second image that can also be expressed as a function of $\boldsymbol{\omega}$.

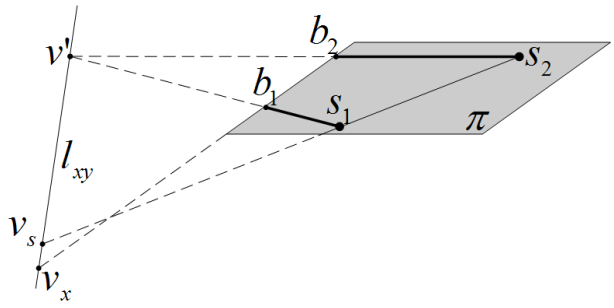


Figure 4.2: The inter-image constraints (4.8, 4.10) can be applied to any corresponding ground lines, excepting shadow lines cast by vertical objects, between two views.

One interesting observation is that the inter-image constraints (4.8, 4.10) can also be applied to any corresponding ground lines between two views, except for the shadow lines cast by the vertical objects, since the concurrent point $\mathbf{v}'$ of all shadow lines on the ground plane has already been enforced in Equation (4.4). For example, we have the image correspondences, $\mathbf{s}_1\mathbf{s}_2$ and $\mathbf{s}'_1\mathbf{s}'_2$, of the 3D line $\mathbf{S}_1\mathbf{S}_2$ between two views, and thus have the inter-image constraints on the vanishing point $\mathbf{v}_s$, which is the intersection of the line $\mathbf{s}_1\mathbf{s}_2$ and the horizon line $\mathbf{l}_{xy}$ (Figure 4.2),

$$\mathbf{v}_s \sim [\mathbf{s}_1 \times \mathbf{s}_2]_\times \boldsymbol{\omega} \mathbf{v}_z. \tag{4.11}$$

Note that the constraint (4.9) typically does not apply to vanishing points other than $\mathbf{v}_x$, since the vanishing line $\mathbf{l}_{xy}$ intersects the vertical plane $\mathbf{H}_{\boldsymbol{\pi}_1}$ only at $\mathbf{v}_x$.

Consequently, we obtain the correct solution by minimizing the following symmetric transfer errors of the geometric distances and epipolar distances of the vanishing points on the line $\mathbf{l}_{xy}$ that involves elements of $\boldsymbol{\omega}$ as:

$$d_1(\mathbf{v}_x^2, \mathbf{H}_{\boldsymbol{\pi}_1}\mathbf{v}_x^1)^2 + \sum_i \{d_1(\mathbf{v}_i^2, \mathbf{H}_{\boldsymbol{\pi}}\mathbf{v}_i^1)^2 + d_2(\mathbf{v}_i^2, \mathbf{F}\mathbf{v}_i^1)^2\}, \tag{4.12}$$

where $\mathbf{v}_i^2$ is the corresponding vanishing point of $\mathbf{v}_i^1$ in the second image, which can also be expressed as a function of $\boldsymbol{\omega}$, $d_1(\mathbf{x}, \mathbf{y})$ is the geometric image distance between the two homogeneous image points represented by $\mathbf{x}$ and $\mathbf{y}$, and $d_2(\mathbf{x}, \mathbf{l})$ is the geometric image distance from an image point $\mathbf{x}$ to an image line $\mathbf{l}$.

Generally, the inter-image constraints (4.12) provide two constraints on $\boldsymbol{\omega}$. As a result, under the assumption of fixed intrinsic parameters and zero skew, it is possible to calibrate a camera from two or more views of a scene containing two vertical objects and their cast shadows. The assumption of zero skew is reasonable both theoretically and practically. As pointed out in [75], the skew will be zero for most normal cameras and can take non-zero values only in certain unusual instances, e.g. taking an image of an image. This argument also coincides with some previous observations, e.g. [105, 153, 202, 40].

Different from the linear orthogonal constraints (4.4), the inter-image constraints in (4.12) are not linear in the elements of $\boldsymbol{\omega}$, and thus need a non-linear optimization process. The starting point can be obtained as follows. As shown in Figure 4.1, the four points $\mathbf{v}_x$, $\mathbf{b}_2$, $\mathbf{b}_1$, and $\mathbf{a}$, are collinear, and their cross-ratio is hence preserved under the perspective projection. Therefore, we have the following equality between two given images:

$$\{\mathbf{v}_x, \mathbf{b}_2; \mathbf{b}_1, \mathbf{a}\}^1 = \{\mathbf{v}_x, \mathbf{b}_2; \mathbf{b}_1, \mathbf{a}\}^2, \tag{4.13}$$

where $\{\cdot, \cdot; \cdot, \cdot\}^i$ denotes the cross ratio of four points, and the superscripts indicate the images in which the cross ratios are taken. This provides us the third constraint on $\boldsymbol{\omega}$, which can be defined

by a 6D vector with five unknowns:

$$\mathbf{w} \sim [1, \omega_{12}, \omega_{22}, \omega_{13}, \omega_{23}, \omega_{33}]^T, \tag{4.14}$$

where ω_{ij} denotes the element in i^{th} row and j^{th} column of $\boldsymbol{\omega}$ in (3.5). The computation detail of Equation (4.13) is given in Section 4.3.2. If we further assume the camera has zero skew and unit aspect ratio (resulting in $\omega_{12} = 0$ and $\omega_{22} = 1$ as shown in Equation (3.9)), then these three known constraints (two from (4.4) and one from (4.13)) are sufficient to solve for the three unknowns: the focal length, f, and the principal point coordinates u_0 and v_0. Note that Equation (4.13), expanded in Equation (4.26) in Section 4.3.2, is quadratic in terms of the elements of $\boldsymbol{\omega}$, and thus the starting point has two solutions of ω_{13}, ω_{23} and ω_{33}, i.e. a two-fold ambiguity. This ambiguity can be eliminated during the minimization process, where only correct solution would minimize the cost functions (4.12).

4.2.3 Maximum Likelihood Bundle Adjustment

The cost function (4.12) described in last subsection 4.2.2 is not optimal in a statistical sense, since it minimizes only local distances rather than the global geometric distances. We can refine it through the standard maximum likelihood inference.

Let us assume that the noise in the measured image feature positions $\mathbf{x}_j^i$, the coordinates of the j-th point as seen by the i-th camera, is additive and has a Gaussian distribution with zero mean and standard deviation σ. Given n images and m corresponding image points, the maximum

likelihood estimate can be obtained by minimizing the following function:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} d_1(\mathbf{x}_j^i, \hat{\mathbf{K}}[\hat{\mathbf{R}}_i \mid \hat{\mathbf{t}}_i]\hat{\mathbf{X}}_j)^2. \tag{4.15}$$

That is, the parameters $\hat{\mathbf{K}}$, $\hat{\mathbf{R}}_i$, $\hat{\mathbf{t}}_i$ and $\hat{\mathbf{X}}_j$ which minimize the sum of the squared geometric image distances, $d_1(\cdot,\cdot)$, between the measured feature locations $\mathbf{x}_j^i$ and the true image points for all points across all views. The minimum of this non-linear cost function, generically termed bundle adjustment in the computer vision and photogrammetry communities, is sought using a Levenberg-Marquardt algorithm [168]. It requires an initial guess of the camera parameters and the 3D world points, which can be obtained using the approach described in the last section and optimal triangulation [74]. Note that the bundle adjustment is efficient only when enough image-to-image corresponding points are available.

4.2.4 Light Source Orientation Estimation

After calibrating the cameras, one would have no difficulty estimating the light source position and orientation provided that the corresponding feature points along the lighting direction can be identified from two views. For example, we can compute the orientation of the light source in 3D by using the optimal triangulation method as follows. Without loss of generality, one can choose the world coordinate frame of the scene shown in Figure 4.1 as follows: origin at $\mathbf{B}_2$, Z-axis along the line $\mathbf{B}_2\mathbf{T}_2$ with the positive direction towards $\mathbf{T}_2$, X-axis along the line $\mathbf{B}_1\mathbf{B}_2$ with the negative direction towards $\mathbf{B}_1$, and the Y-axis given by the right-hand rule. Then, using the triangulation

method [74], it is possible to compute the 3D locations of $\mathbf{B}_1$ and $\mathbf{S}_1$. Consequently, the orientation of the light source is given by $\mathbf{n} = \mathbf{B}_1 - \mathbf{S}_1$.

Alternatively, since in our case the light source is far away, we only need to measure the orientation of the light source, which can be expressed by two angles: the polar angle ϕ between the lighting direction and the Z-axis (the vertical direction), and the azimuthal angle θ in the ground plane from the X-axis (the line $\mathbf{B}_1\mathbf{B}_2$). Note that the imaged light source $\mathbf{v}$ is the vanishing point along lighting direction since $\mathbf{v}$ intersects the plane at infinity $\boldsymbol{\pi}_\infty$, and also that $\mathbf{v}'$ is the projection of $\mathbf{v}$ on ground plane (the X-Y plane). Consequently, the polar angle ϕ with the vertical Z-axis and the azimuthal angle θ can be measured from the corresponding vanishing points using

$$\phi = \cos^{-1} \frac{\mathbf{v}_z^T \boldsymbol{\omega} \mathbf{v}}{\sqrt{\mathbf{v}^T \boldsymbol{\omega} \mathbf{v}} \sqrt{\mathbf{v}_z^T \boldsymbol{\omega} \mathbf{v}_z}}, \tag{4.16}$$

$$\theta = \cos^{-1} \frac{\mathbf{v}_x^T \boldsymbol{\omega} \mathbf{v}'}{\sqrt{\mathbf{v}'^T \boldsymbol{\omega} \mathbf{v}'} \sqrt{\mathbf{v}_x^T \boldsymbol{\omega} \mathbf{v}_x}}, \tag{4.17}$$

where $\mathbf{v}$ can be computed as,

$$\mathbf{v} \sim (\mathbf{t}_1 \times \mathbf{s}_1) \times (\mathbf{t}_2 \times \mathbf{s}_2). \tag{4.18}$$

In our experiments, we used the second method to compute ϕ and θ for each view and the shown results are the averaged ones.

4.2.5 Algorithm Outline

The complete algorithm will now be summarized. The input is a pair of images of a scene containing two vertical objects and their shadows on the ground plane cast by an infinite light (e.g. the

sun). Both vertical objects and shadow lines should be visible in both images. The output are the camera parameters and the orientation of the light source. A top-level outline of the image is as follows.

1. Identify a seed set of image-to-image matches between the two images. At least six points, $\hat{\mathbf{t}}_i$, $\hat{\mathbf{b}}_i$ and $\hat{\mathbf{s}}_i$ ($i = 1, 2$), are needed, though more are preferable. The feature points can be extracted as shown in Section 4.3.1.

2. Obtain close-form solution

   (a) Compute $\mathbf{v}_z$ (4.1) and $\mathbf{v}'$ (4.2) for each view, and thus obtain two linear constraints on the $\boldsymbol{\omega}$ (4.4).

   (b) Compute the third constraint (4.13) as described in Section 4.3.2.

   (c) Solve two linear constraints (4.4) and one quadratic constraint (4.13) by assuming $\boldsymbol{\omega}_{12} = 0$ and $\boldsymbol{\omega}_{22} = 1$. This gives us $\boldsymbol{\omega}_{13}$, $\boldsymbol{\omega}_{23}$ and $\boldsymbol{\omega}_{33}$ up to an ambiguity.

   (d) Compute the fundamental matrix $\mathbf{F}$ (when enough point correspondences are available) and the planar homographies $\mathbf{H}_{\boldsymbol{\pi}}$ and $\mathbf{H}_{\boldsymbol{\pi}_1}$ as described in Section 4.2.2.

   (e) Minimize (4.12) using the above $\boldsymbol{\omega}_{13}$, $\boldsymbol{\omega}_{23}$ and $\boldsymbol{\omega}_{33}$ as starting points.

   (f) Compute the camera internal parameters as shown in 3.6 and the camera external parameters as described in [40].

3. Use Bundle Adjustment (4.15) to refine the close-form solution.

4. Compute the orientation of the light source (4.16,4.17).

4.3 Implementation Details

4.3.1 Feature Extraction

Features, extracted either automatically (e.g. using an edge or corner detector) or interactively, are subject to errors. The features are mostly the image locations of the top, $\mathbf{t}_i$, and the base, $\mathbf{b}_i$ of the vertical objects, and the shadow positions $\mathbf{s}_i$ of the top locations $\mathbf{t}_i$ ($i = 1, 2$). In this implementation, we use a maximum likelihood estimation (MLE) method similar to [39] with the uncertainties in the locations of points $\mathbf{t}_i$, $\mathbf{b}_i$ and $\mathbf{s}_i$ modeled by the covariance matrices $\mathbf{\Lambda}_{\mathbf{t}_i}$, $\mathbf{\Lambda}_{\mathbf{b}_i}$ and $\mathbf{\Lambda}_{\mathbf{s}_i}$ respectively.

In the error-free case, imaged shadow relations (illuminated by a point light source) are modeled by a *planar homology* as introduced in Section 3.5.2 (see Figure 3.4 and also Figure 4.3). One important property of a planar homology is that the corresponding lines, i.e. lines through pairs of corresponding points, intersect on the axis: for example, for the lines $\mathbf{t}_1\mathbf{t}_2$ and $\mathbf{s}_1\mathbf{s}_2$,

$$\mathbf{b}_1\mathbf{b}_2 \cdot (\mathbf{t}_1\mathbf{t}_2 \times \mathbf{s}_1\mathbf{s}_2) = 0. \tag{4.19}$$

Another important property of a planar homology is that the cross ratio defined by the vertex, $\mathbf{v}$, the corresponding points, $\mathbf{t}_i$ and $\mathbf{s}_i$, and the intersection, $\mathbf{i}_i$, of their join $\mathbf{t}_i\mathbf{s}_i$ with the axis, is the characteristic invariant of the homology, and are the same for all corresponding points,

$$\{\mathbf{v}, \mathbf{t}_1; \mathbf{s}_1, \mathbf{i}_1\} = \{\mathbf{v}, \mathbf{t}_2; \mathbf{s}_2, \mathbf{i}_2\}. \tag{4.20}$$

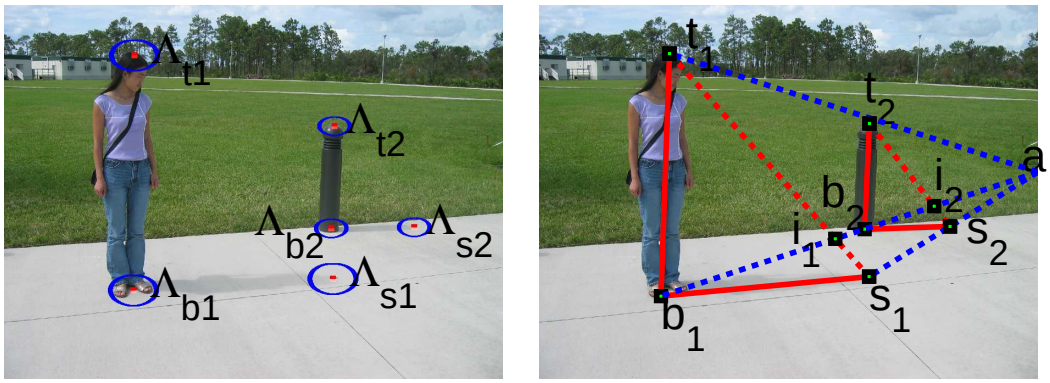


Figure 4.3: Maximum likelihood estimation of the feature points: (left) The uncertainty ellipses are shown. These ellipses are specified by the user, and indicate the confidence region for localizing the points. (right) Estimated feature points are satisfying the alignment constraints (4.19) and (4.20).

Therefore, we determine the maximum likelihood estimates of the features' true locations ($\hat{\mathbf{t}}_i$, $\hat{\mathbf{b}}_i$ and $\hat{\mathbf{s}}_i$) by minimizing the sum of the Mahalanobis distances between the input points and their MLE,

$$\arg\min_{\hat{\mathbf{x}}_j} \sum_{j=1,2} \sum_{\mathbf{x}=\mathbf{t},\mathbf{b},\mathbf{s}} (\mathbf{x}_j - \hat{\mathbf{x}}_j)^T \mathbf{\Lambda}_{\mathbf{x}_j}^{-1} (\mathbf{x}_j - \hat{\mathbf{x}}_j), \tag{4.21}$$

subject to the alignment constraints (4.19) and (4.20).

The covariance matrices $\mathbf{\Lambda}_{\mathbf{t}_i}$, $\mathbf{\Lambda}_{\mathbf{b}_i}$ and $\mathbf{\Lambda}_{\mathbf{s}_i}$ are not necessarily isotropic or equal. For example, in our experiments for the image shown in Figure 4.3, the second view of the first real image set shown in Figure 4.9, we set $\mathbf{\Lambda}_{\mathbf{t}_1} = \mathbf{\Lambda}_{\mathbf{b}_1} = diag([20^2, 10^2])$ and $\mathbf{\Lambda}_{\mathbf{t}_2} = \mathbf{\Lambda}_{\mathbf{b}_2} = \mathbf{\Lambda}_{\mathbf{s}_1} = \mathbf{\Lambda}_{\mathbf{s}_1} = 8^2\mathbf{I}_{2\times 2}$. The image is of size 2272×1704.

Once the six points, $\mathbf{t}_i$, $\mathbf{b}_i$ and $\mathbf{s}_i$, are interactively identified in one view, we first initialize the positions of feature points in extra views using the wide baseline matching [195] and then refine

them using the same MLE estimates (4.21). In our implementation, we assume the uncertainties in the locations of points $\mathbf{t}_i$, $\mathbf{b}_i$ and $\mathbf{s}_i$ are similar from view to view, and hence we use the same $\mathbf{\Lambda}_{\mathbf{t}_i}$, $\mathbf{\Lambda}_{\mathbf{b}_i}$ and $\mathbf{\Lambda}_{\mathbf{s}_i}$ for all images.

Note that, to obtain more robust solutions, we also compute point correspondences other than the six feature points $\mathbf{t}_i$, $\mathbf{b}_i$ and $\mathbf{s}_i$, $i = 1, 2$, for the computation of inter-image planar homographies and fundamental matrix in Equation (4.12).

4.3.2 Computation Details of the Cross Ratio Constraint

This section provides the computation details on the cross ratio constraint in Equation (4.13). Denoting the computed homogeneous image line $\mathbf{b}_1\mathbf{b}_2$ as $[l_1,\ l_2,\ l_3]^T$, and the vanishing point $\mathbf{v}_z$ as $[v_{z_1},\ v_{z_2},\ v_{z_3}]^T$, $\mathbf{v}_x$ in Equation (4.7) can be expressed as

$$\mathbf{v}_x = [\mathbf{D}_1^T\mathbf{w},\ \mathbf{D}_2^T\mathbf{w},\ \mathbf{D}_3^T\mathbf{w}]^T, \tag{4.22}$$

where $\mathbf{w}$ is given in Equation (4.14), and $\mathbf{D}_i$, $(i = 1, 2, 3)$, are defined as

$$\mathbf{D}_1 = [0,\ l_3v_{z_1},\ l_3v_{z_2},\ -l_2v_{z_1},\ l_3v_{z_3} - l_2v_{z_2},\ -l_2v_{z_3}]^T, \tag{4.23}$$

$$\mathbf{D}_2 = [-l_3v_{z_1},\ -l_3v_{z_2},\ 0,\ l_1v_{z_1} - l_3v_{z_3},\ l_1v_{z_2},\ l_1v_{z_3}]^T, \tag{4.24}$$

$$\mathbf{D}_3 = [l_2v_{z_1},\ l_2v_{z_2} - l_1v_{z_1},\ -l_1v_{z_2},\ l_2v_{z_3},\ -l_1v_{z_3},\ 0]^T. \tag{4.25}$$

Now we are ready to compute the cross ratio involving $\mathbf{v}_x$ in Equation (4.13), i.e. a ratio of ratios of lengths on the line $\mathbf{b}_1\mathbf{b}_2$. Since cross ratio is invariant to the projective coordinate frame chosen

for the line, we can use the coordinate differences of the four points $(\mathbf{v}_x, \mathbf{b}_2, \mathbf{b}_1, \mathbf{a})$ rather than their lengths to express the cross ratio to simplify the constraints on $\boldsymbol{\omega}$. In our implementation, we choose x-coordinates of image points due to the fact that imaged lines $\mathbf{b}_1\mathbf{b}_2$ in our cases are more horizontal then vertical, i.e. the absolute value of the line's slope is less than one. As a result, Equation (4.13) can be expanded as

$$\{\mathbf{v}_x, \mathbf{b}_2; \mathbf{b}_1, \mathbf{a}\} = \frac{(\mathbf{v}_x - \mathbf{b}_2)(\mathbf{b}_1 - \mathbf{a})}{(\mathbf{v}_x - \mathbf{a})(\mathbf{b}_1 - \mathbf{b}_2)} = \frac{\mathbf{D}_4^T \mathbf{w}}{\mathbf{D}_5^T \mathbf{w}}, \tag{4.26}$$

with

$$\mathbf{D}_4 = (b_{1_x} - a_x)(\mathbf{D}_1 - b_{2_x}\mathbf{D}_3),$$

$$\mathbf{D}_5 = (b_{1_x} - b_{2_x})(\mathbf{D}_1 - a_x\mathbf{D}_3),$$

where b_{1_x}, b_{2_x} and a_x are the x-coordinates of $\mathbf{b}_1, \mathbf{b}_2$ and $\mathbf{a}$ respectively.

4.4 Degenerate Configurations

The proposed algorithm, like almost any algorithm, has degenerate configurations where the parameters can not be estimated. It is important to be aware of these configurations to obtain reliable results, in practice, by avoiding them.

Generally, as an approach based on vanishing points, the method is degenerate if the vanishing points along some directions are at infinity. Geometrically, vanishing points go to infinity when the perspective transformation degenerates to affine transformation, where parallel lines in 3D remain parallel in the image plane, and when the vanishing point direction is parallel to the image plane since the vanishing point of a line is obtained by intersecting the image plane with a ray parallel

to the world line and passing through the camera center. Note that here parallelism to the image plane means the parallelism in 3-space. Algebraically, a vanishing point goes to infinity means that the third component of the homogeneous vanishing point coordinates equals to zero. In practice, if conditions are near degenerate then the solution is ill-conditioned, and the particular member of the family of solutions is dominated by "noise". For calibration algorithms using the vanishing points, therefore, it is always important to work with images with relatively large projective distortion when it is possible.

The method mainly employs the orthogonality relationship (4.4) of the vanishing points, $\mathbf{v}_z$ and $\mathbf{v}'$, of two perpendicular 3D world directions, and the inter-image constraints (4.12, 4.13) associated with vanishing points other than $\mathbf{v}'$ on the vanishing line $\mathbf{l}_{xy}$ as shown in Figure 4.1. Therefore, it is mainly important to verify the vanishing point, $\mathbf{v}_z$, along the vertical direction, and other vanishing points of world lines that are perpendicular to the vertical direction, i.e. vanishing points on the horizon line $\mathbf{l}_{xy}$.

The rest of this section is organized as follows. First, possible singularities with geometrical explanations are derived. Then, we especially give algebraic analysis on the situations where the inter-image constraints fail. For this analysis, the vanishing point $\mathbf{v}_x$ is used as an example and, however, the derivation can be easily extended to others.

4.4.1 Vanishing Point at Infinity

Liebowitz and Zisserman [105] demonstrate an easy but efficient method to identify the degeneracies that occur when constraints on the ω are not independent, which exploits the linearity of the equations in terms of the elements of ω. Although we introduce new constraints for camera calibration in Equation (4.12,4.13), the price to pay is increased difficulty in identifying degenerate configurations. Although it is still possible to apply the approach [105] to check the dependeny between the linear constraints in Equation (4.4) that arise from the parallel property of shadows cast by infinite light source, we would like to derive all possible singularities with geometrical explanations.

The following analyses are most related to Sturm and Maybank [153] but different mainly in three aspects. First, although both their and our approaches utilize the orthogonality constraints in equation (4.4), they have also the constraints arising from the normal property of $\mathbf{r}_1$ and $\mathbf{r}_2$ as described in Section 3.7 (originally used by Zhang in [200]), while we derive the new inter-image constraints in Equations (4.13,4.12). Notice that in their case the use of the normal properties of $\mathbf{r}_1$ and $\mathbf{r}_2$ is viable due to the fact that they have the world to image planar homography. Second, the two planes π and π_1 (See Figure 4.1) in our case are perpendicular to each other, which is a special case of those addressed in [153]. Third, only a subset of their degenerate cases apply to our configuration.

Our method degenerates in the following cases (parallelism and perpendicularity here are on 3D domain): (I) the image plane is parallel to plane π_1, (II) the image plane is parallel to the

ground plane $\boldsymbol{\pi}$, and (III) the image plane is perpendicular to both $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}$. In case I, both $\mathbf{v}_x$ and $\mathbf{v}_z$ go to infinity. In case II, all the vanishing points on the line $\mathbf{l}_{xy}$ will be infinite. Case II is different from case I in that the principal point can be recovered in case II, but not in case I. In case II, the vanishing points on the ground plane $\boldsymbol{\pi}$, e.g. $\mathbf{v}_x$ and $\mathbf{v}'$, are the vanishing points for directions parallel to the image plane. Therefore, the vertical direction, which is perpendicular to the ground plane $\boldsymbol{\pi}$, must be orthogonal to the image plane, and parallel to the principal axis of the camera. The vanishing point of this principal axis is nothing but the principal point, which is thus given by the finite vanishing point $\mathbf{v}_z$. The same derivation is valid for case I only if $\mathbf{v}'$ is orthogonal to $\mathbf{v}_x$, which is not necessarily true. Note that unlike [153], the aspect ratio λ can not be recovered in our case when one plane is parallel to the image plane, since we have no Euclidean information about the 3D world coordinates. In case III, if the degenerate situation happens for both views, we are able to intersect the two lines $\mathbf{b}_1\mathbf{b}_2$ in two views to obtain the principal point as analyzed above. These analyses are partially validated in Section 5.4.1.

4.4.2 Difficulty in Computing $\mathbf{v}_x$

One important step of the proposed algorithm is to express $\mathbf{v}_x$ as a function of $\boldsymbol{\omega}$ in equation (4.7). Therefore, the method degenerates when $\mathbf{v}_x$ can not be computed from equation (4.7). In other words, lines $\mathbf{b}_1\mathbf{b}_2$ and $\mathbf{l}_{xy}$ are parallel to each other in the image plane. To simplify the analysis, we adopt the world coordinate definition in Section 4.2.4, although the algebraic derivation is independent of the definition of the world coordinate.

Line $\mathbf{b}_1\mathbf{b}_2$ is the imaged projection of the 3D X-axis and, therefore, passes through the image of the world origin $[0\ 0\ 0\ 1]^T$, which in our case is

$$\mathbf{b}_2 \sim [\mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3\ \mathbf{p}_4][0\ 0\ 0\ 1]^T = \mathbf{p}_4. \tag{4.27}$$

It also passes through the vanishing point $\mathbf{v}_x \sim \mathbf{p}_1$ (shown in (3.11)) along the 3D X-axis direction $[1\ \ 0\ \ 0]^T$. Consequently, line $\mathbf{b}_1\mathbf{b}_2$ can be expressed as $\mathbf{b}_1\mathbf{b}_2 \sim \mathbf{p}_1 \times \mathbf{p}_4$. Similarly, line $\mathbf{l}_{xy}$ is the horizon line of the ground plane (X-Y plane), and therefore passes through two vanishing points $\mathbf{v}_x$ and $\mathbf{v}_y(\sim \mathbf{p}_2)$, $\mathbf{l}_{xy} \sim \mathbf{p}_1 \times \mathbf{p}_2$. When lines $\mathbf{b}_1\mathbf{b}_2$ and $\mathbf{l}_{xy}$ are parallel to each other, it indicates that the two lines intersect at infinity, i.e. the third component of the imaged intersection, $\mathbf{b}_1\mathbf{b}_2 \times \mathbf{l}_{xy}$, equals to zero. By using the property that the cofactor matrix is related to the way matrices distribute with respect to the cross product [67], one can obtain

$$\begin{aligned}
\{\mathbf{b}_1\mathbf{b}_2 \times \mathbf{l}_{xy}\}_3 = 0 &\iff \{(\mathbf{p}_1 \times \mathbf{p}_4) \times (\mathbf{p}_1 \times \mathbf{p}_2)\}_3 = 0 \\
&\iff \{[\mathbf{K}^*(\mathbf{r}_1 \times \mathbf{RC})] \times [\mathbf{K}^*(\mathbf{r}_1 \times \mathbf{r}_2)]\}_3 = 0 \\
&\iff \{\mathbf{K}[(\mathbf{r}_1 \times \mathbf{RC}) \times \mathbf{r}_3]\}_3 = 0 \\
&\iff \{\mathbf{K}[r_2]_\times^{-1}\mathbf{K}^T\mathbf{K}^{-T}(R\tilde{t} \times r_3)\}_3 = 0 \\
&\iff \{\mathbf{K}[([\mathbf{r}_1]_\times[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3]\mathbf{C}) \times \mathbf{r}_3]\}_3 = 0 \\
&\iff \{\mathbf{K}(C_y\mathbf{r}_3 - C_z\mathbf{r}_2) \times \mathbf{r}_3\}_3 = 0 \\
&\iff \{C_z\mathbf{K}\mathbf{r}_1\}_3 = 0 \\
&\iff C_z = 0 \ or\ \{\mathbf{v}_x\}_3 = 0,
\end{aligned} \tag{4.28}$$

where $\mathbf{K}^*$ denotes the matrix of cofactors of $\mathbf{K}$, and $\{\mathbf{z}\}_3$ denotes the 3^{rd} element of the vector $\mathbf{z}$. Algebraically, $C_z = 0$ implies line $\mathbf{b}_1\mathbf{b}_2$ coincide with the vanishing line $\mathbf{l}_{xy}$, in which case the

pole-polar relationship in equation (4.3) offers two independent constraints on the camera internal parameters per view. The two constraints might be used in camera calibration as shown in [106]. However, the configuration $C_z = 0$ indicates that the camera center lies on the ground plane $\boldsymbol{\pi}$ in Figure 4.1, which is unlikely to happen in practice since it is rare to capture images or videos like that. Practically, therefore, it is only necessary to avoid the case $\{\mathbf{v}_x\}_3 = 0$, where $\mathbf{v}_x$ goes to infinity.

4.5 Experimental Results

The proposed method aims to directly calibrate cameras for applications where it is difficult to calibrate cameras beforehand using special calibration patterns with known geometry, and when the numbers of views is insufficient to employ self-calibration methods. This experiments, therefore, focus on the cases where only two views are available.

First, synthetic data is used to examine the performance of the proposed method with respect to the following cases:

1. different noise levels varying from 0.3 pixels to 3.0 pixels,
2. different relative orientation (parallel to perpendicular) between the vertical plane passing through the two vertical lines and the image plane,
3. different errors on the orhtogonality constraint, i.e. the angles between the vertical objects and the ground plane are varying (from 0.3 degrees to 3.0 degrees in the experiments).

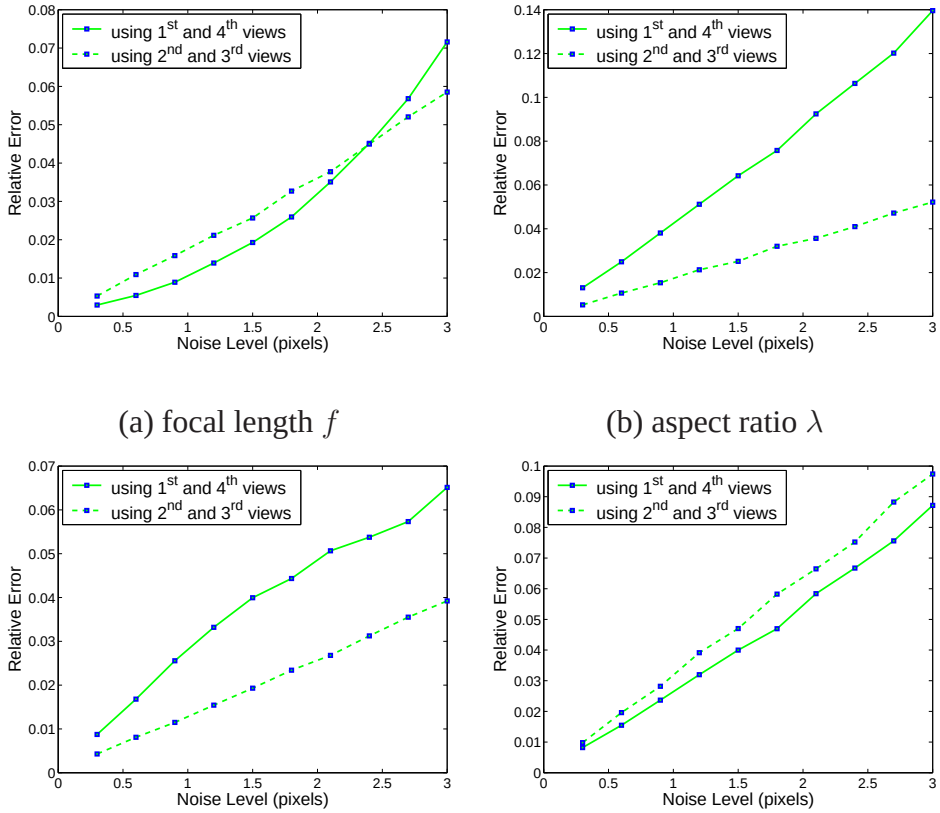
Table 4.1: External Parameters of four different viewpoints.

View	camera position	"at" position
1^{st}	(10, -100+random(1), 40)	(-100, 0, 0)
2^{nd}	(-150, -100+random(1), 40)	(-100, 0, 0)
3^{rd}	(10, -100+random(1), 100)	(-100, 0, 0)
4^{th}	(-150, -100+random(1), 100)	(-100, 0, 0)

Second, we compare the performance of our method with the ground truth using the real images.

4.5.1 Computer Simulation

The simulated camera has the focal length of $f = 1000$, the aspect ratio of $\lambda = 1.06$, the skew of $\gamma = 0$, and the principal point at $u_0 = 8$ and $v_0 = 6$. The synthetic light source is at infinity with the polar angle $\phi = \arctan(0.5)$ degrees and the azimuthal angle $\theta = 60$ degrees. The two vertical objects have lengths 100 and 80 units respectively, and the distance between the two vertical objects is 75 units. To approximate the real cases where there exist some other point correspondences besides the vertical line segments and their cast shadows, we generated another ten 3D points randomly located on a hemisphere above the ground plane, centered at $(75, 0, 0)$ with radius 80. In the experiments presented herein, we generated four views with camera and "look at" positions listed in Table 4.1, where we follow the camera (eye) coordinate specification in OpenGL fashion. Therefore, $"at" - camera$ is the principal view direction.

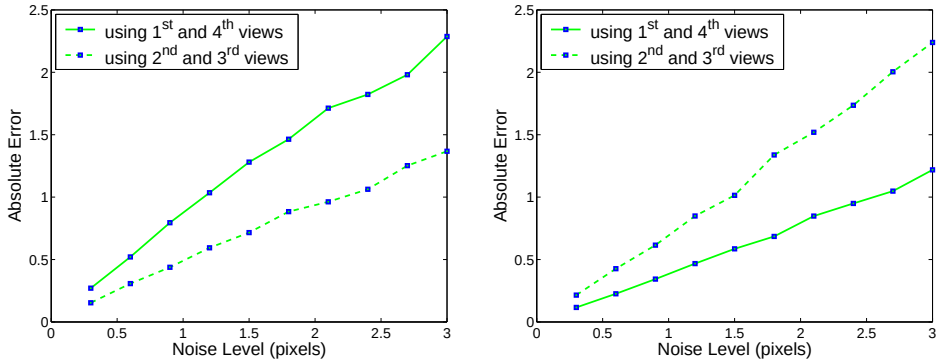


(a) focal length f (b) aspect ratio λ

(c) u_0 of principal point (d) v_0 of principal point

Figure 4.4: The performance of our method on the camera calibration under different noise levels.

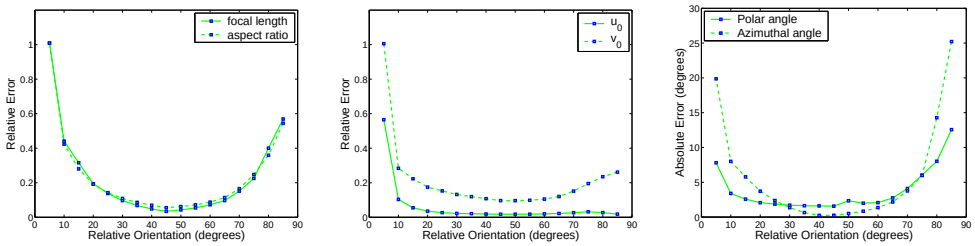
Performance versus noises: In this experimentation, we used two combinations of image pairs (views) in Table 4.1. The first combination composes of the 1^{st} and 4^{th} views, while the second one includes the 2^{nd} and 3^{rd} views. Gaussian noise with zero mean and a standard deviation of $\sigma \leq 3.0$ pixels was added to the projected image points. The estimated camera parameters were then compared with the ground truth. For each noise level, we perform 1000 independent trials, and the final averaged results of camera internal parameters are shown in Figure 4.4, while Figure 4.5 illustrates the performance of our method on the light source orientation estimation. As argued



(a) Polar angle ϕ (b) Azimuthal angle θ

Figure 4.5: The performance of our method on the light source orientation estimation under different noise levels.

by [173, 202], the relative difference with respect to the focal length rather than the absolute error is a more geometrically meaningful error measure for the camera internal parameters. Therefore, we measured the relative error of f, u_0 and v_0 with respect to true f while varying the noise level from 0.3 pixels to 3.0 pixels. For the aspect ratio λ, we measure the relative error with respect to itself. Evidently, errors increase almost linearly with respect to the noise level for both the camera internal parameters and the light source orientations. For $\sigma = 1.5$, which is comparable to the typical noise in practical calibration, the relative error of focal length f is 1.93% for the first combination and 2.57% for the second combination. When we add more noise, the relative errors of focal lengthes keep increasing until it reaches 7.16% for the first combination and 5.85% for the second one when $\sigma = 3.0$. The maximum relative error of aspect ratio is 13.96% for the first combination, 5.22% for the second combination, while the maximum relative errors of principal

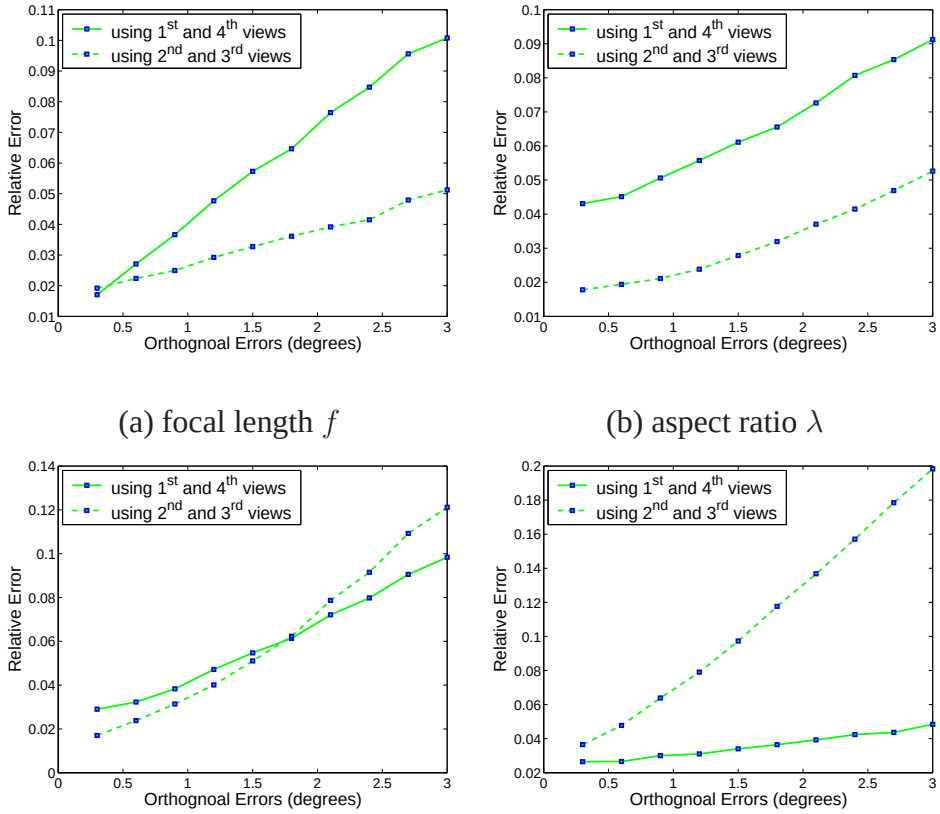


(a) (b) (c)

Figure 4.6: The performance of our method with respect to the relative orientation between the vertical plane passing through the two vertical lines and the image plane. (a) The focal length and the aspect ratio. (b) The principal point. (c) The light source orientation.

points are around 6.51% for u_0 and about 9.74% for v_0. Finally, the computed orientation of the light source is within $\pm 2.5^\circ$ for both polar and azimuthal angles.

Performance versus the orientation of the image plane. This experiment examines the influence of the relative orientation between the vertical plane passing through the two vertical lines and the image plane. Two views are used. The orientation of the image planes of two views are chosen as follows: the image planes are initially parallel to the vertical plane; a rotation axis is randomly chosen from a uniform sphere; the image planes are then rotated around that axis with angle ψ. Considering the fact that extracted feature points will in practice be affected by noise, we also add a typical noise level of $\sigma = 1.0$ pixels to all projected image points. We repeated this process 1000 times and computed the average errors. The angle ψ varies from 5° to 85°, and the results are shown in Figure 4.6. Several important observations are evident. First, the results partially validate the analysis of degenerate cases in the Section 4.4.1. In case I where the image plane is parallel to the vertical plane, i.e. $\psi = 0^\circ$, our method degenerates. In case III, i.e. $\psi = 90^\circ$, our method is

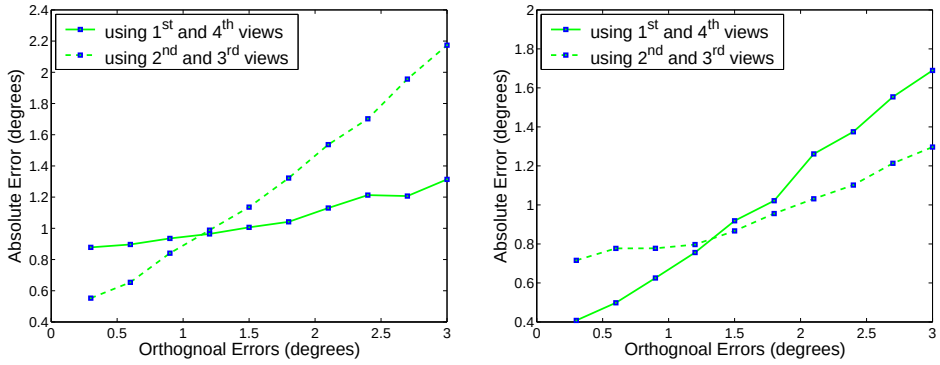


(a) focal length f (b) aspect ratio λ

(c) u_0 of principal point (d) v_0 of principal point

Figure 4.7: The performance of our method on the camera calibration with respect to the orthogonality errors. The orthogonality errors here means the errors in the angles between the vertical objects and the ground plane, which are assumed to be $90°$.

also degenerate, but is able to recover the principal point since in our experiments the image planes of both views are also approximately perpendicular to the ground plane. The second observation is that the best performance seems to be achieved with an angle around $45°$. Third, the minima of the curve is sufficiently broad such that acceptable results can be found in practice from wide range



(a) Polar angle ϕ (b) Azimuthal angle θ

Figure 4.8: The performance of our method on the light source orientation estimation with respect to the orthogonality errors.

of views. Note that in practice, when ψ increases, foreshortening makes the feature detection less precise, which is not considered in this experiment.

Performance versus orthogonality error: The last experiment is carried out to evaluate how sensitive the algorithm is to errors in the angles β between the vertical objects and the ground plane, which are assumed to be $90°$. For the angle β of each vertical object, independent Gaussian noise with zero mean and a standard deviation of $\sigma \leq 3.0$ degrees was added. Considering the fact that extracted feature points will in practice be affected by noise, we also added a typical noise level of $\sigma = 1.0$ pixels to all projected image points. We repeat this process 1000 times, and the final averaged results of camera calibration are shown in Figure 4.7, while Figure 4.8 illustrates the performance of our method on the light source orientation estimation. The errors in both camera parameters and light source orientation increase almost linearly with respect to the orthogonality errors. Notice also that the errors do not go to zero as orthogonality errors go towards zero due to

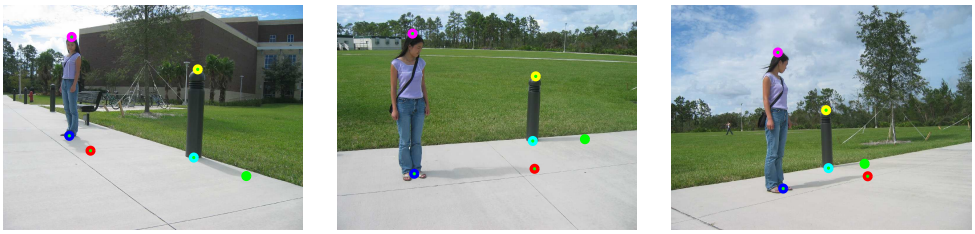


Figure 4.9: Three images of a standing person and a lamp. The circle marks in the images are the minimal data computed by the method described in Section 4.3.

Table 4.2: Calibration results for the first real image set.

Error	**Image Pair**			**Ground Truth**
	(1,2)	(1,3)	(2,3)	[105]
f (relative error)	3200.8 (1.44%)	3137.1 (-0.58%)	3211.1 (1.77%)	3155.3
λf (relative error)	3206.1 (-3.21%)	3171.0 (-4.27%)	3216.4 (-2.90%)	3312.5
u_0 (relative error)	1172.4 (0.28%)	1334.9 (5.43%)	1170.1 (0.21%)	1163.6
v_0 (relative error)	895.8 (-0.57%)	902.7 (-0.35%)	901.5 (-0.39%)	913.8

the added noise in image projections. For example, the relative errors of focal lengths are around 1.8% under orthogonality error $\sigma = 0.3^\circ$ and pixel noise $\sigma = 1.0$ pixels (Figure 4.7 (a)), while the focal length errors are around 1.4% when there is the same pixel noise $\sigma = 1.0$ pixels but no orthogonality error (Figure 4.4 (a)).

4.5.2 Real Data

We also applied our method to real images. The image set consists of three views of a standing person and a lamp, which provided two vertical lines for camera calibration (see Figure 4.9). For each pair of images, we applied our algorithm independently, and the results are shown in Table 4.2. To evaluate our results, we obtained a least-squares solution for internal parameters of a non-natural camera from over-determined noisy measurements, i.e. five images with three mutually orthogonal vanishing points per view, using the constraints described in [105]. We compared our results to those listed in the last column in Table 4.2. The largest relative error of the focal length, in our case, is less than 4.5%. The maximum relative error of principal point is around 5%. In addition, the computed polar angle ϕ and azimuthal angle θ are 44.54 and 33.22 degrees respectively, while they are 45.09 and 32.97 degrees by using the camera intrinsic parameters in the last column of Table 4.2. The errors could be attributed to several sources. Besides noise, non-linear distortion and imprecision of the extracted features, one source is the casual experimental setup using minimal information, which is deliberately targeted for a wide spectrum of applications. Despite all these factors, our experimentations indicate that the proposed algorithm provides good results.

4.6 Applications

To validate our calibration techniques, we present three practical applications. To further demonstrate the efficiency and applicability of the inter-image constraints, we first apply them to symmet-

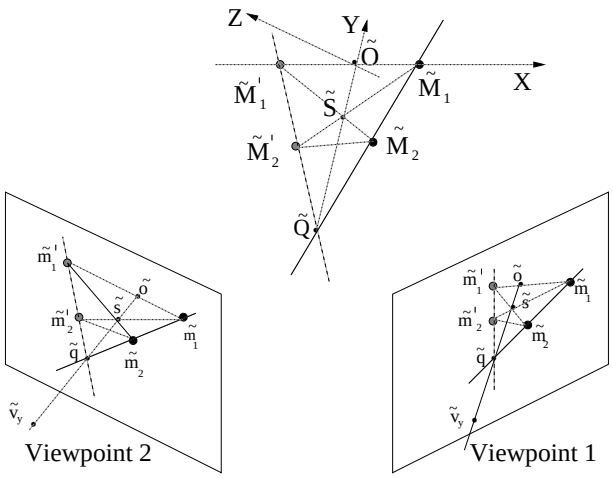


Figure 4.10: Four feature points forming an isosceles trapezoid in the world plane. The plane of symmetry is defined by the z-axis and the line $\mathbf{OQ}$. The points $\mathbf{M}_1$ and $\mathbf{M}_2$ are two arbitrary feature points on the calibration object, and $\mathbf{M}'_1$ and $\mathbf{M}'_2$ are their symmetric counterparts. Alternatively, $\mathbf{M}_1$ and $\mathbf{M}_2$ are two feature points on a line through the point $\mathbf{Q}$ and $\mathbf{M}'_1$ and $\mathbf{M}'_2$ are their position after pivoting the line around the point $\mathbf{Q}$.

ric and 1D objects, which is especially useful in calibration of a large number of cameras [202]. The second application is the reconstruction of partially viewed symmetric objects, and the last one is the image based metrology.

4.6.1 Calibration Using Symmetric and 1D Objects

In the case that the cameras are observing an object that has a plane of symmetry $\mathbf{\Pi}$, for each object point $\mathbf{M}$, there exists a point $\mathbf{M}'$ on the object, which is symmetrically situated with respect to $\mathbf{\Pi}$.

Any such pair of symmetric points will then define a 3D line $\mathbf{M}'\mathbf{M}$, which is perpendicular to the plane of symmetry. Our method assumes that two such pairs are viewed by a camera. Denote the first pair by $\mathbf{M}_1$ and $\mathbf{M}'_1$, and the second pair by $\mathbf{M}_2$ and $\mathbf{M}'_2$. Clearly, the two lines $\mathbf{M}'_1\mathbf{M}_1$ and $\mathbf{M}'_2\mathbf{M}_2$ are coplanar and parallel, and in general yield an isosceles trapezoid as shown in Figure 4.10. Therefore, the vanishing point $\mathbf{v}_x$ along the direction perpendicular to the symmetric plane $\mathbf{\Pi}$ is known, and is given by

$$\mathbf{v}_x = (\mathbf{m}_1 \times \mathbf{m}'_1) \times (\mathbf{m}_2 \times \mathbf{m}'_2). \tag{4.29}$$

Similar to the work [186], the intersection between $\mathbf{\Pi}$ and the plane passing the two lines $\mathbf{m}'_1\mathbf{m}_1$ and $\mathbf{m}'_2\mathbf{m}_2$ can be uniquely defined as the line,

$$\mathbf{l}_s = \mathbf{s} \times \mathbf{q}, \tag{4.30}$$

where $\mathbf{s}$ is the intersection of $\mathbf{m}'_1\mathbf{m}_2$ and $\mathbf{m}'_2\mathbf{m}_1$, and $\mathbf{q}$ is the intersection of $\mathbf{m}_1\mathbf{m}_2$ and $\mathbf{m}'_1\mathbf{m}'_2$. However, the main difference is that the pole-polar relationship with respect to $\mathbf{v}_x$ and $\mathbf{l}_s$ does not hold in our configuration as shown in section 4.6.1.1.

It is interesting to note that this configuration of coplanar points may also be viewed as two separate configurations of collinear points in the 3D space. In other words, if we take the lines $\mathbf{M}_1\mathbf{M}_2$ and $\mathbf{M}'_1\mathbf{M}'_2$ as two different positions of a 1D object pivoting around the point $\mathbf{Q}$, then we can also apply the technique proposed herein to the images of this 1D object. The only difference now is that the set of four coplanar points $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}'_1$, and $\mathbf{M}'_2$, are captured in two separate images. Therefore, the number of images required in the 1D case is twice as many as the number of images required in the 2D case, i.e. a minimum of four: one pair from the same viewpoint but

with the 1D object at two distinct positions, and another pair from a different viewpoint but the same two positions of the object that were used in the first pair. Accordingly, two arbitrary points $\mathbf{M}_1$ and $\mathbf{M}_2$ with unknown locations on the 1D object and the corresponding points $\mathbf{M}'_1$ and $\mathbf{M}'_2$ after moving the 1D object to another position (with $\mathbf{Q}$ kept fixed), yield one isosceles trapezoid $\mathbf{M}'_1\mathbf{M}'_2\mathbf{M}_2\mathbf{M}_1$. We call this a viewpoint of the isosceles trapezoid.

In the coordinate frame described above, the vanishing point $\mathbf{v}_x$ and the line $\mathbf{l}_s$ play the same roles as $\mathbf{v}_z$ and the line $\mathbf{b}_1\mathbf{b}_2$ in the Figure 4.1, and therefore the inter-image constraints (4.12) can be applied with the starting point similarly computed using (4.13). However, no orthogonality constraint in equation (4.4) is available, since only one vanishing point is known. As a compensation, therefore, we assume a unit aspect ratio and zero camera skew, in which case the camera calibration matrix $\mathbf{K}$ in Equation (3.1) will reduce to

$$\mathbf{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.31}$$

Below, we use an alternative method as detailed in [29, 26] to calibrate the camera. The basic idea is that both the camera internal and external parameters can be expressed as functions of the principal point $\mathbf{c} = (u_0, v_0)$ and, therefore, we formulate our problem in terms of the inter-image homography that minimizes the symmetric transfer error of geometric distances,

$$(f, \mathbf{c}) = \arg\min_{\mathbf{\Gamma}} \sum_i d(\mathbf{m}_i, \mathbf{H}_{f,\mathbf{c}}^{-1}\mathbf{m}'_i))^2 + d(\mathbf{m}'_i, \mathbf{H}_{f,\mathbf{c}}\mathbf{m}_i)^2 \tag{4.32}$$

where $d(\cdot,\cdot)$ is the Euclidean distance between the image points, $\mathbf{\Gamma}$ is the 2D search space of the solution for $\mathbf{c}$, $\mathbf{H}_{f,\mathbf{c}} = \hat{\mathbf{H}}'_w\hat{\mathbf{H}}_w^{-1}$ is the inter-image homography (which is only a function of $\mathbf{c}$).

Table 4.3: External Parameters for seven different viewpoint.

Viewpoints	θ_x	θ_y	θ_z	t_x	t_y	t_z
1^{st}	10	-7	3.8	20	40	350
2^{nd}	12	6	-5	20	-24	350
3^{rd}	12	13	-12	10	70	360
4^{th}	12	30	-12	10	20	350
5^{th}	12	16	-5	10	-40	360
6^{th}	10	5	3.8	20	26	370
7^{th}	12	16	-15	10	-4	380

In the implementation, we take advantage of the fact that the principal points of recent CCD cameras are very close to the center of the image [75]. Therefore, the search space $\mathbf{\Gamma}$ that minimize the cost function in (4.32) is then narrowed down to a 2D window around the image center. The solution is therefore found without resorting to non-linear minimization techniques, i.e. by sampling the solution space within the 2D search window.

The proposed approach was tested on an extensive set of simulated and real data. Below, we first show the synthetic simulations for the 1D object (the 2D case is similar). We then show the results for real images. We used 1D objects with similar configurations as those in [201] for comparison. The simulated camera has a focal length of $f = 1020$, unit aspect ratio, zero skew, and the principal point at $(316,\ 243)$. The image resolution is 640×480. We observed the 1D objects randomly at seven positions listed in Table 4.3. For each observation, we switched the

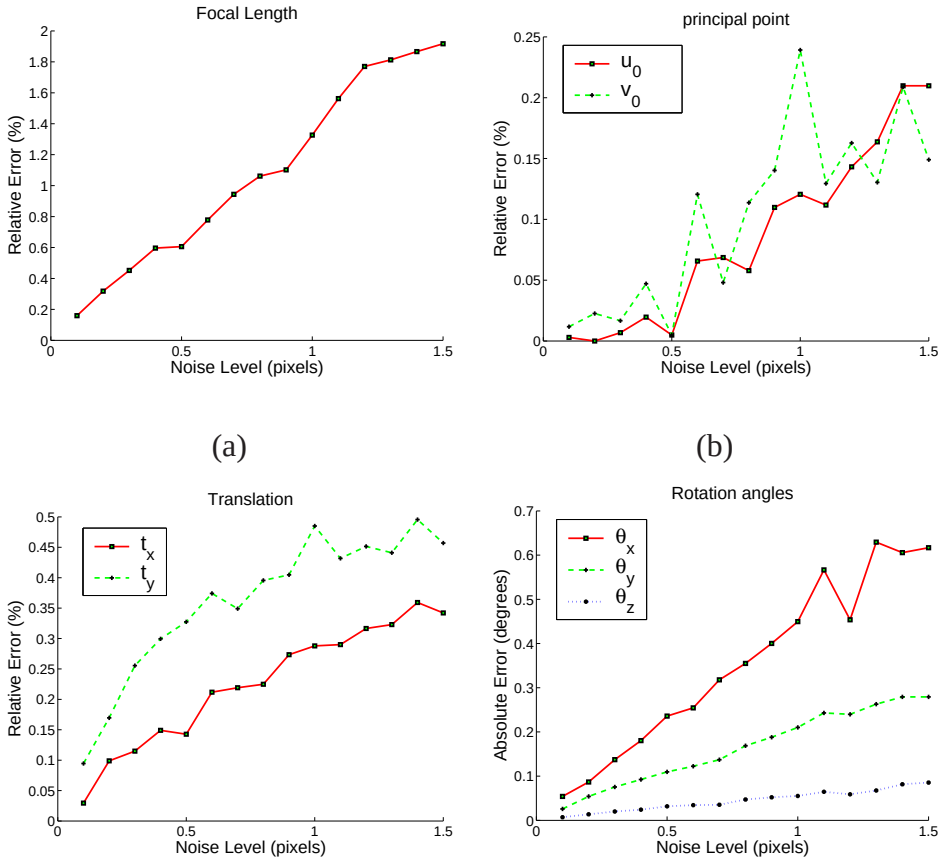
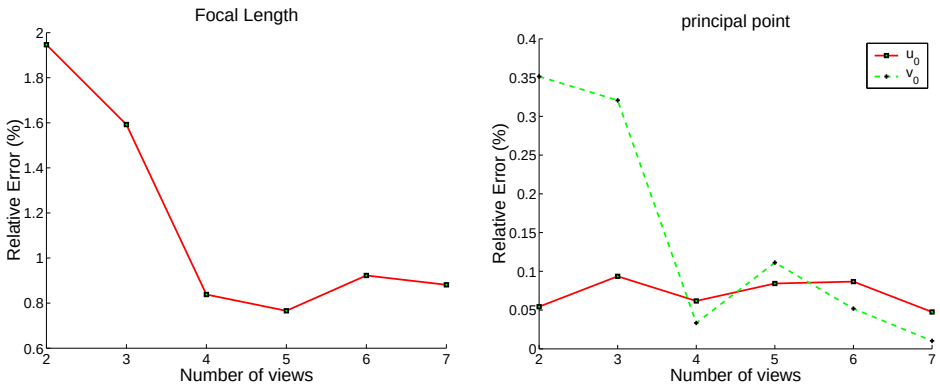


Figure 4.11: Performance vs noise (in pixels) averaged over 100 independent trials: (a), (b) and (c) relative error for f, principal point and translation, (d) absolute errors in and rotation angles.

1D object between two positions of the 3D points $\mathbf{M}_1 = [-75\ 0\ 0]^T$, $\mathbf{M}_2 = [-45\ \ -150\ 0]^T$, $\mathbf{M}_1' = [75\ 0\ 0]^T$, $\mathbf{M}_2 = [45\ \ -150\ 0]^T$. The 2D searching space is 15×15 with 1.0 pixel interval in each direction. Of course, once we get close enough to the solution, we can also refine further the search window, if we wish higher accuracy.

Performance Versus Noise Level: In this experimentation, we used the first five image pairs (viewpoints) in Table 4.3. The estimated camera intrinsic and extrinsic parameters were then

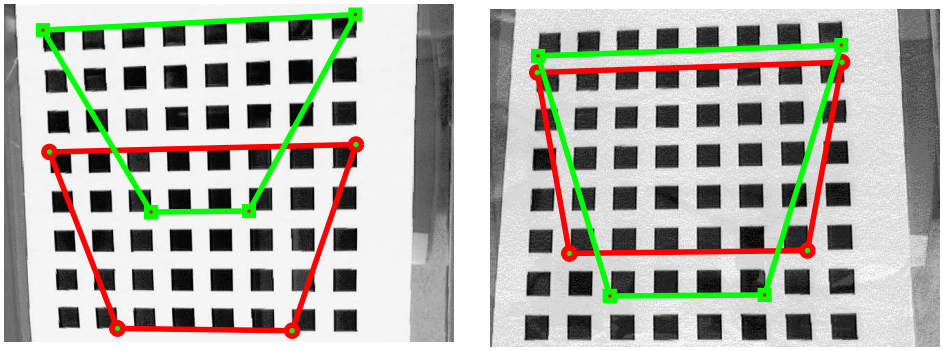


(a) (b)

Figure 4.12: (a) & (b) Performance vs number of viewpoints averaged over 100 independent trials: (a) relative error for f, (b) absolute errors in principal point.

compared with the ground truth, while adding a zero-mean Gaussian noise varying from 0.1 pixels to 1.5 pixels. Results are shown in Figure 4.11. For noise level of 1.5 pixels, which is larger than the typical noise in practical calibration [200], the relative error of the focal length f is 1.92%. The maximum relative error of principal points is around 0.24%. Excellent performance is achieved for all extrinsic parameters, i.e. relative errors less than 0.36% for t_x and 0.5% for t_y, absolute errors less than 0.63 degree for θ_x, less than 0.28 degree for θ_y and less than 0.086 degree for θ_z. At 1 pixel noise level, in our 1D algorithm, the relative errors of the focal length and the principal point do not exceed 1.56%.

Performance Versus Number of Viewpoints: We also examined the performance with respect to the number of viewpoints (i.e. the number of image pairs). We varied the number of available viewpoints from 2 to 7. Results are shown in Figure 4.12. For these set of experimentations the noise level was kept at 1.5 pixels, and the results were again averaged over 100 independent trials.



(a) (b)

Figure 4.13: Four trapezoids projected into two images.

Figure 4.14: Three collinear points along a TV antenna.

For four or more viewpoints, the relative error of f drops sharply to an average of 0.85%, and the relative errors of u_0 and v_0 drop sharply to an average of 0.71% and 0.52%, respectively. The more viewpoints we have, the more accurate camera calibration will be in practice.

For real data, we applied our method to both 1D and 2D objects, and compared our algorithm to Zhang's flexible calibration method [200]. The comparison is meant to be only informal, since our camera model is simpler (but not requiring 3D-2D correspondences). For 2D calibration, we used the data set used by Zhang with no knowledge about the 3D coordinates of the calibration grid. The

Table 4.4: Results for real data compared to Zhang's results.

quadruple	(1234)	(1235)	(1245)	(1345)	(2345)	mean	dev
α [200]	831.81	832.09	837.53	829.69	833.14	832.85	2.90
β [200]	831.82	832.10	837.53	829.91	833.11	832.90	2.84
f **(Ours)**	**833.55**	**834.29**	**835.96**	**832.33**	**834.28**	**834.08**	**1.32**
u_0 [200]	304.53	304.32	304.57	303.95	303.53	304.18	0.44
u_0 **(Ours)**	**303.50**	**304.25**	**303.25**	**304.00**	**304.50**	**303.90**	**0.52**
v_0 [200]	206.79	206.23	207.30	207.16	206.33	206.76	0.48
v_0 **(Ours)**	**217.25**	**213.25**	**212.25**	**205.25**	**208.50**	**211.30**	**4.60**

radial distortion was removed according to his experimental results. We used four trapezoids (i.e. 16 corners) as shown in Figure 4.13 for gathering the error statistics. To evaluate our results, we also used an approach similar to [200] based on estimating the uncertainty of the results using the standard deviation of the estimated internal parameters f, and (u_0, v_0). We evaluated the variation of calibration results among all quadruples of images. Results are compared to those by Zhang in Table 4.4.

For the 1D object, we used the antenna of a home TV and took 16 images with 8 different viewpoints. One pair of images observed from the same viewpoint is shown in Figure 4.14. Three points along the antenna are chosen to generate isosceles trapezoids. The calculated intrinsic parameters using these images are listed in Table 4.5. We used the sample mean as the ground truth

Table 4.5: Intrinsic parameters for real data

image set	f	rel. err.	u_0	rel. err.	v_0	rel. err.
(1 2 3 4 5 6 7)	2443.22	-0.61%	1137.00	0.12%	838.67	-0.02%
(1 2 3 4 5 6 8)	2604.29	5.94%	1143.00	0.36%	844.00	0.20%
(1 2 3 4 5 7 8)	2510.88	2.14%	1125.33	-0.36%	838.00	-0.05%
(1 2 3 4 6 7 8)	2347.97	-4.48%	1137.83	0.15%	838.00	-0.05%
(1 2 3 5 6 7 8)	2472.31	0.57%	1131.00	-0.13%	838.00	-0.05%
(1 2 4 5 6 7 8)	2411.91	-1.88%	1137.00	0.12%	838.00	-0.05%
(1 3 4 5 6 7 8)	2456.22	-0.08%	1131.00	-0.13%	840.67	0.06%
(2 3 4 5 6 7 8)	2418.83	-1.60%	1131.00	-0.13%	838.00	-0.05%

and also show the relative difference with respect to the mean value of f. The largest relative distance of f, in our case, is less than 6%.

4.6.1.1 *Relation to Existing Methods*

The proposed camera calibration approach using symmetric objects is related to but different from the work in [186] using surfaces of revolution. In [186], Wong et al. use the line $\mathbf{l}_s$ corresponding to the projection of the axis of revolution in the image plane for calibration. In their case, this line is the vanishing line of the plane containing the camera center and the axis of revolution. By choosing their $\mathbf{v}_x$ as the vanishing point along the normal to this plane, they obtain a pole-polar relationship

between $\mathbf{v}_x$ and $\mathbf{l}_s$, i.e. $\mathbf{l}_s = \omega\mathbf{v}_x$ (see [186] for derivation). This pole-polar relationship provides two linear constraints per image. Therefore Wong et al. can solve for the intrinsic parameters using two images. In our case, however, as depicted in Figure 4.6.1.1, $\mathbf{v}_x$, is not the vanishing point along the normal of the plain $\mathbf{\Pi}_s$ containing the camera center and the line $\mathbf{SQ}$. Therefore it does not have a pole-polar relationship with the vanishing line $\mathbf{l}$ of that plane.

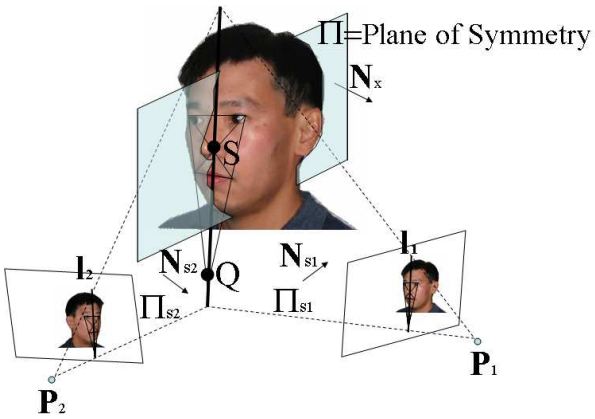


Figure 4.15: Our configuration: the plane Π_s that defines the image line $\mathbf{l}$, is different from the plane of symmetry Π, whose normal $\mathbf{N}_x$ defines the vanishing point $\mathbf{v}_x$. As a result, in our configuration, $\mathbf{l} \neq \omega\mathbf{v}_x$.

Formally, in our configuration

$$\mathbf{v}_x = \mathbf{P}\mathbf{N}_x \tag{4.33}$$

$$\text{and} \quad \mathbf{N}_x = \mathbf{\Omega}\mathbf{\Pi} \tag{4.34}$$

where $\mathbf{\Omega}$ denotes the absolute dual quadric [75]. But clearly, as can be seen in Figure 4.6.1.1, in general back-projecting the line $\mathbf{l}$ would not yield the plane of symmetry $\mathbf{\Pi}$, i.e.

$$\mathbf{\Pi} \neq \mathbf{P}^T \mathbf{l} \tag{4.35}$$

From (4.33), (4.34), and (4.35), we deduce that in our case

$$\mathbf{l} \neq \boldsymbol{\omega} \mathbf{v}_x \tag{4.36}$$

Therefore, we do not have a pole-polar relationship between the image line $\mathbf{l}$ and the vanishing point $\mathbf{v}_x$.

4.6.2 Reconstruction of Partially Viewed Symmetric Objects

This section demonstrates the application of the proposed method on 3D Euclidean reconstruction of a partially viewed symmetric object. We use a two-step method to build the 3D model of a scene given two images of a symmetric object [50]. The first step is direct camera calibration as described in Section 4.6.1. The second step is an *extended triangulation*, which exploits symmetry property to handle points that are occluded or are not seen in one image. Symmetry, which is a property shared by many natural and man-made objects, is a rich source of information in images. Methods that exploit symmetry to impose constraints on the 3D structure of the scene have been occasionally explored in the past [117, 206]. Francois et. al. [7] have presented a method for reconstructing mirror symmetric scenes from a single view by synthesizing a second camera based on the first one. However, they assume that their cameras are positioned in a restricted mirror

Table 4.6: Estimated 3D coordinates at 1.0 pixel noise level

Points	true X	Est. X	true Y	Est. Y	true Z	Est. Z
1^{st}	-85	-82.47	0	0.002	0	-0.127
2^{nd}	-125	-121.72	-150	-144.57	0	-0.037
3^{rd}	125	121.44	-150	-144.31	0	-0.455
4^{th}	85	82.13	0	0.133	0	-0.552
5^{th}	-85	-83.43	0	0.298	-100	-95.73
$6\&7^{th}$	± 125	± 124.37	-100	-96.30	-115.69	-109.70
8^{th}	85	81.98	0	0.577	-100	-95.93

symmetric setup. As a result the calibration is fixed by the restricted geometric configuration of the cameras, and hence is assumed known. We describe a more general framework, where unknown cameras can view a symmetric object from most positions and orientations, except for the degenerate configurations discussed in Section 4.4.

Assume that we have found the projection matrices of the two cameras, say $\mathbf{P}_1$ and $\mathbf{P}_2$, using the method described in Section 4.6.1. We can use the optimal triangulation method [74] to extract the 3D coordinates from point correspondences. Triangulation for the commonly assumed case, where the correspondences are visible in both images is well known. However, in real scenarios, we may face a situation where one point $\mathbf{M} = [X\ Y\ Z\ 1]^T$ is visible in one image and not in the second one (or vice versa). For a symmetric object, however, its symmetric counterpart, i.e. the

Table 4.7: Performance vs noise (in pixels)

Noise	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.
0.2	1.60	0.78	1.48	1.77	1.80	1.88
0.4	1.44	0.51	1.24	1.40	1.43	1.60
0.6	1.76	0.37	1.37	1.60	1.46	1.76
0.8	2.14	0.66	2.51	2.82	2.86	2.92
1.0	3.30	0.83	2.95	3.48	3.10	3.14
1.2	1.97	0.92	1.94	2.33	2.88	2.55
1.5	2.70	0.65	2.87	2.83	2.58	2.77

point $\mathbf{M}'$, might be visible, in which case we can show that the 3D coordinates of these points can still be recovered unambiguously.

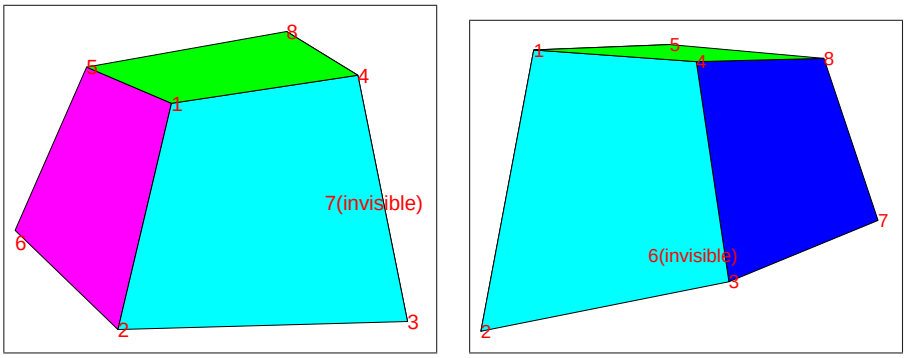
In the world coordinate frame, the relationship between the two symmetric points $\mathbf{M}$ and $\mathbf{M}'$ (symmetric with respect to the world Y-Z plane) is given by:

$$\mathbf{M}' = \mathbf{D}\mathbf{M} \tag{4.37}$$

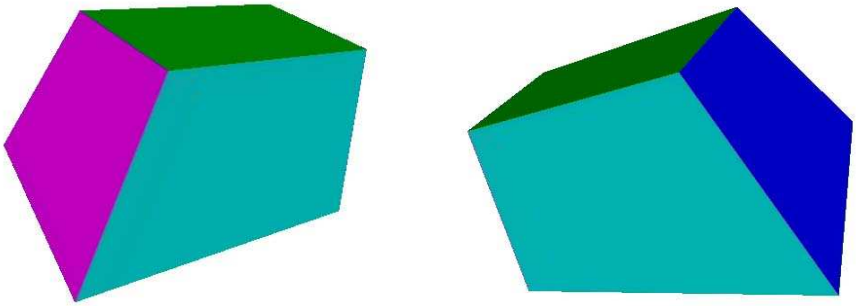
where $\mathbf{D} = \text{diag}(-1, , 1, 1, 1)$. Thus the image point $\mathbf{m}'$ is given by

$$\mathbf{m}' = \mathbf{P}_2\mathbf{M}' = \mathbf{P}_2\mathbf{D}\mathbf{M} \tag{4.38}$$

In other words, image points $\mathbf{m}$ and $\mathbf{m}'$ of two symmetric points may be viewed as the projections of the same 3D world point $\mathbf{M}$ by projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2\mathbf{D}$. Therefore, 3D reconstruction



(a) (b)



(c) (d)

Figure 4.16: Synthetic results for 1.0 pixel noise level: (a) & (b) input image pair, (c) & (d) Two snapshots of the reconstructed textured model.

can be achieved for regions that are viewed by only one camera (e.g. due to occlusions or partial view), using the symmetric part of the object.

The proposed approach has been tested on both simulated and real data. For the synthetic data shown herein the image resolution was 480×320. Eight points with 3D coordinates shown in Table 4.6 were reconstructed to recover four planes as shown in Figure 4.16. Note that the point pair 6^{th} and 7^{th} in Table 4.6 are not both visible in the images. The 7^{th} point is not seen in the left image, while the 6^{th} point can not be seen in the right image. Hence, they only differ in their

(a) (b)

(c) (d)

Figure 4.17: Reconstruction example of partially viewed symmetric objects. (a) & (b) Two real images of a partially viewed symmetric object, (c) & (d) snapshots of the reconstructed 3D model including the occluded left and right portions

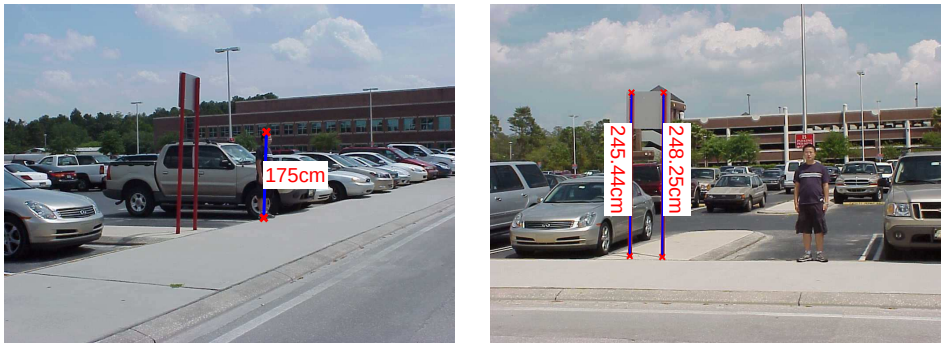
x-coordinates by a sign. In the experiments presented herein, Gaussian noise with zero mean and a standard deviation of $\sigma \in [0, 0.5]$ was added to the projected image points. For each noise level, we ran our algorithm independently 100 times, and all the results shown are the averaged ones. The reconstructed 3D points were compared with the ground truth. In 1.0 pixel noise level, the results are shown in Table 4.6. To evaluate the performance with respect to noise, we also measured the

absolute errors of X, Y and Z for every points. Mean and standard deviation of all coordinates are shown in Table 4.7 with different noise levels. Finally, two snapshots of the resulting textured model are shown in Figure 4.16 together with the synthetic input images.

We experimented with various real objects that contained symmetric structures. Experimental results were verified against ground truth, which indicate an excellent performance for our approach, with the standard deviation of errors in the recovered distance ratios under 3.5. Figure 4.17 shows the reconstructed VRML model from a pair of real images of a symmetric object. Note that the left side and the right side are only visible in one image. However, our technique accurately recovers both sides as shown in the snapshots of the reconstructed 3D model shown in Figure 4.17, (c) and (d).

4.6.3 Image Based Metrology

In [37], Criminisi proposed an approach for single view metrology, and showed that affine scene structure may be recovered from a single image without any prior knowledge of the camera calibration parameters. The limitation of their approach is that they require that three mutually orthogonal vanishing points to be available simultaneously in the image plane. Also, to recover the metric measurements they require three reference distances. Their advantage however is that they need only one image to solve the problem. In this dissertation, we propose two new methods on metrology.



(a) (b)

Figure 4.18: Measuring height of vertical objects: (a) The standing person has known height, (b) Computed heights of two sign board posts.

(a) (b)

Figure 4.19: Measurements which might be difficult in practice.

The first approach requires two vertical objects, and therefore only one vanishing point along a vertical to a reference plane. However this method would require two images to solve the problem with only one reference distance. Examples of images where such scenario may apply are commonly encountered in indoor and outdoor environments, where there is a ground plane and some up-right objects, e.g. humans, street lamps, trees, etc., but not all vanishing points available, see

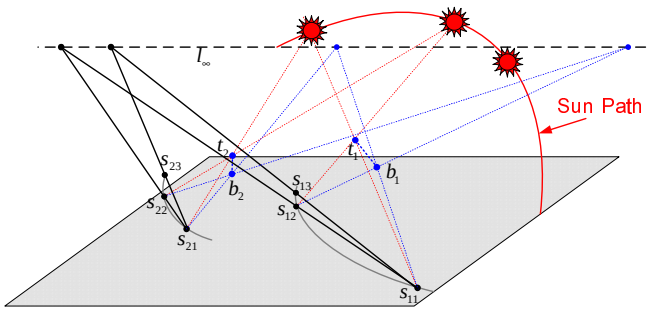


Figure 4.20: The 3D points $\mathbf{t}_i$ cast shadows at different time j at positions $\mathbf{s}_{ij}$.

for instance Figure 4.18. Note also that Criminisi et al. [37] can only perform measurements in the reference plane and the planes parallel to it. In our approach, we can directly perform measurements outside the reference plane and along non-parallel lines. The details of this method are described in [28, 51], and this section only demonstrates the results shown in 4.18 and 4.19, where we used the height of the standing person as the reference. In Figure 4.18, the computed heights of sign board posts are similar, which coincide with the ground truth. To test the accuracy of our algorithm, we also compared the computed results with ground truth measurements. For instance, in Figure 4.19, the estimated stick's height is 100.75cm, while the ground truth is 99.4cm. The distance between the bottom of the standing person and bottom of the stick is 116cm, the estimated one is 119.47cm. The approach can be used to measure heights of the objects that are not accessible for direct measurement too. For instance, we estimated the tree's height as 263.62cm. Other estimated distances which might be difficult to measure in practice are also shown.

The second method requires no presence of objects, but only shadows on the ground plane cast by two unknown stationary 3D points. We first utilize the tracked shadow positions to compute the horizon line. This is based on our observation that any two 3D corresponding lines on the shadow

trajectories are parallel to each other, and therefore the imaged lines provide the vanishing points along directions perpendicular to the vertical point, which can be fit from detected vertical objects. The basic idea is illustrated in Fig. 4.20. Suppose $\mathbf{t}_i$ is a 3D point and $\mathbf{b}_i$ is the closest point on the ground plane to $\mathbf{t}_i$. Therefore, line $\mathbf{t}_i\mathbf{b}_i$ is perpendicular to the ground plane. $\mathbf{s}_{ij}$ is the cast shadow point in different time j. Because the sun is so far away (approximately 1.52×10^{12} meters) from the earth, it is reasonable to consider the sunlight as parallel light rays. Therefore, line pair $\mathbf{b}_i\mathbf{s}_{ij}$ and $\mathbf{b}_{i'}\mathbf{s}_{i'j}$ are parallel. We can also have

$$\frac{\mathbf{b}_1\mathbf{s}_{11}}{\mathbf{b}_2\mathbf{s}_{21}} = \frac{\mathbf{b}_1\mathbf{s}_{12}}{\mathbf{b}_2\mathbf{s}_{12}} = \frac{\mathbf{t}_1\mathbf{b}_1}{\mathbf{t}_2\mathbf{b}_2}, \tag{4.39}$$

Note that this equation is true in 3D, not in the image plane. Consequently, the two world triangles $\triangle\mathbf{b}_1\mathbf{s}_{11}\mathbf{s}_{12}$ and $\triangle\mathbf{b}_2\mathbf{s}_{21}\mathbf{s}_{22}$ are similar, and that the world line $\mathbf{s}_{11}\mathbf{s}_{12}$ and $\mathbf{s}_{21}\mathbf{s}_{22}$ are parallel to each other. Therefore, the shadows observed over time is sufficient to provide the horizon line $\mathbf{l}_\infty$, for which the object top and bottom points ($\mathbf{t}_i$, $\mathbf{b}_i$) are not required to present in the image. If we assume the vertical vanishing point is also available using methods [89] and [106], the computed horizon line, together with the vertical vanishing point, provide the pole-polar relationship on camera calibration. In this section, we are only interested in the measurements of the relative heights of the two objects $\mathbf{t}_i\mathbf{b}_i$ and $\mathbf{t}_j\mathbf{b}_j$.

First we recover the affine property of the ground plane by a projective transformation (or affine rectification):

$$\mathbf{H}_p = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{pmatrix}, \tag{4.40}$$

where $[l_1 \ \ l_2 \ \ l_3]^T$ is the vanishing line $\mathbf{l}_\infty$. As a result, the affine properties such as the ratio of lengths on parallel lines are invariant. From equation (4.39) and the fact that the two world triangles $\triangle \mathbf{b}_1\mathbf{s}_{11}\mathbf{s}_{12}$ and $\triangle \mathbf{b}_2\mathbf{s}_{21}\mathbf{s}_{22}$ are similar, we have the equivalence

$$\frac{\mathbf{s}_{ik}\mathbf{s}_{il}}{\mathbf{s}_{jk}\mathbf{s}_{jl}} = \frac{\mathbf{t}_i\mathbf{b}_i}{\mathbf{t}_j\mathbf{b}_j}, \ \ \forall i, j, k, l. \tag{4.41}$$

Therefore, given n observations of shadows of two object i and j, we have C_n^2 solutions of $\frac{\mathbf{t}_i\mathbf{b}_i}{\mathbf{t}_j\mathbf{b}_j}$, from which the optimal solution can be computed as a weighted mean. The weight is dependent on the relative distance between the k and l.

4.7 Conclusion

The proposed novel inter-image constraints relax some of the limitations in other calibration algorithms using vanishing points, and extend the current state-of-the-art to situations where only one vanishing point is known. These constraints were applied to objects with mirror symmetry, 1D objects, and vertical objects and shadows which are frequently found in both indoor and outdoor environments, e.g. a chair, a vase, eye glasses, a face, a car, an airplane, an antenna, a street pole, standing humans etc. Therefore, this work has also the contribution in the benefit of increasing the accessibility of the calibration objects.

The fact that prior knowledge of the 3D coordinates of the calibration object is not required, makes the method a versatile tool that can be used without requiring a precisely machined calibration rig (e.g. grids), and also makes calibration possible when the object is not accessible for

measurements, e.g. in remote sensing, image-based rendering [78, 40, 43], or simply when the images are taken by other people, e.g. Zhang's real data. These flexibilities make the method an off-the-shelf and easy-to-use tool available to a wide spectrum of users. Experimental results show that the method provides very promising solutions even with minimum information. Applications on 3D reconstruction and image based metrology demonstrate the usability and efficiency of the proposed method.

CHAPTER 5

SELF-CALIBRATION USING CONSTANT MOTION

The previous chapter describes a new technique for calibration using novel constraints derived from the scene geometry. In this chapter, we propose an alternative approach exploiting the motion of the camera itself. Ultimately, as described later in this thesis, both approaches will be used to facilitate video post-production under various practical scenarios. Below, we investigate using constant inter-frame motion for self-calibration from an image sequence of an object rotating around a single axis with varying camera internal parameters. Our approach makes use of the facts that in many commercial systems rotation angles are often controlled by an electromechanical system, and that the inter-frame essential matrices are invariant if the rotation angles are constant but not necessarily known. Therefore, recovering camera internal parameters is possible by making use of the equivalence of essential matrices, which relate the unknown calibration matrices to the fundamental matrices computed from the point correspondences. This chapter also describes a linear method that works under restrictive conditions on camera internal parameters, the solution of which can be used as the starting point for an iterative non-linear method with looser constraints. The results are refined by enforcing the global constraint that the projected trajectory of any 3D point should be a conic after compensating for the focusing and zooming effects. Finally, using

the bundle adjustment method tailored to the special case, i.e. static camera and constant object rotation, the 3D structure of the object is recovered and the camera parameters are further refined simultaneously. To determine the accuracy and the robustness of the proposed algorithm, the results on both synthetic and real sequences are also presented in this chapter.

5.1 Introduction

Acquiring 3D models from circular motion sequences, particularly turn-table sequences, has been widely used by computer vision and graphics researchers, e.g. [164, 124, 17, 155], since these methods are simple and robust. Generally, the whole reconstruction procedure includes: first, the determination of camera positions at different viewpoints or, equivalently, the different positions of the rotating device; second, the detection of object boundaries or silhouettes; third, the extraction of a visual hull as the surface model from a volume representation [91]. Fitzgibbon et. al. [52] extended the analysis of the circular motion to recover unknown rotation angles from uncalibrated image sequences based on a projective geometry approach and multi-view geometric constraints. Mendonça et. al. [118, 119] recovered the circular motion by using surface profiles. Wong et. al. [186] also presented a method for camera calibration using surfaces of revolution, which is related to circular motion since an object placed on a turn-table spans a surface of revolution. Recently, Jiang et. al. developed new methods to compute single axis motion by either fitting the conic to the locus of the tracked points in at least five images [81] or computing a plane homography from a minimal of two points in four images [79]. Colombo et. al. [23] improved the approach [186] in

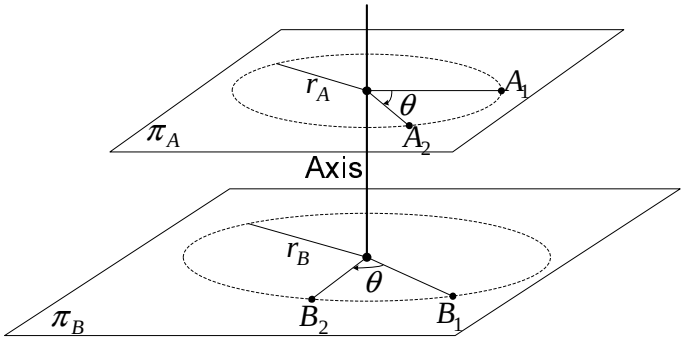
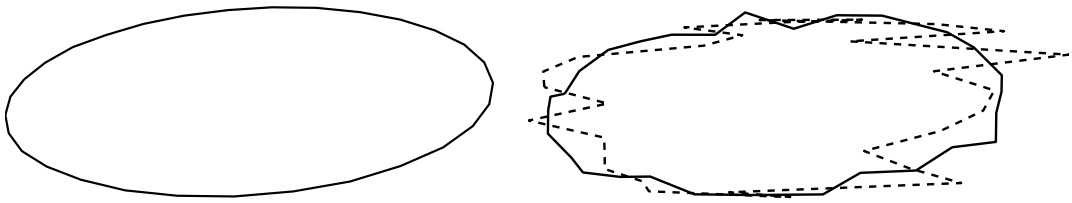


Figure 5.1: The geometric configuration of a single axis rotation in 3D space. The space points $\mathbf{A}_i$ and $\mathbf{B}_i$ are circularly moving around the fixed rotation axis on two different planes $\boldsymbol{\pi}_A$ and $\boldsymbol{\pi}_B$. To aid in visualization, we assume that the rotation axis is vertical, so that the 3D points rotate in horizontal planes. In our case, the relative angle between views i and $i+1$ are constant and denoted by θ.

which the calibration of a natural camera, a pin-hole camera with zero skew and unit aspect ratio [105], requires the presence of two different surfaces of revolutions in the same view. In addition, the method [23] relaxes the conditions, claimed by [81], that three ellipses are needed to compute the imaged circular points.

However, most of these methods deal with the case in which a static camera with fixed internal parameters views an object rotating on a turn-table (Figure 5.1), and utilize the fixed image entities of the circular motion. These fixed image entities (Figure 5.3) include two lines: one is the image of the rotation axis $\mathbf{l}_s$, a line of fixed points, while the other one, called the horizon line $\mathbf{l}_\infty$, is the image of the vanishing line of the horizontal planes, e.g. $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$. Unlike the image of the rotation axis, the horizon line is a fixed line, but not a line of fixed points. Under the assumption



(a) (b)

Figure 5.2: The projected trajectories of 3D points under circular motion. Both (a) and (b) are the projected trajectories across 90 views of a typical scene point under the configuration described in Section 5.4.1. Different from (a) which uses a fixed camera, the solid curve in (b) has focal lengths chosen with a mean value of 1000 pixels and a standard deviation $\sigma_f = 10$ pixels, while the dashed curve in (b) has $\sigma_f = 100$ pixels.

of the fixed camera internal parameters, the image of the absolute conic is fixed for a rigid motion. Therefore, there are two points, $\mathbf{i}$ and $\mathbf{j}$, located at the intersection of the absolute conic with the horizon line, that remain fixed in all images. Actually, these two fixed points are the images of the two circular points on the horizontal planes [75], and can be found by the intersections of conic loci of corresponding points since the trajectories of space points are circles in 3D space and intersect in the circular points on the plane at infinity. However, these entities are fixed only when the camera has fixed internal parameters. For example, the projected trajectory of a 3D point is not a conic any more when the camera's internal parameters are varying (Figure 5.2 (b)).

In this chapter we concentrate on the situation where the stationary camera is free to zoom and focus, and assume that in many commercial systems, rotation angles are often controlled by an electromechanical system [164, 124, 155, 115], i.e. they are constant and even known. It

is shown that the inter-frame essential matrices are invariant if the rotation angle is constant but not necessary known and, therefore, recovering camera internal parameters is possible by making use of the equivalence of essential matrices. We also introduce a linear method that works under restrictive conditions on camera internal parameters, such as known camera skew, aspect ratio and principal point, the solution of which can be used as the starting point of an iterative non-linear methods with looser constraints. The results are optimized by enforcing the global constraints that the projected trajectories of 3D points should be conics after compensating the focusing and zooming effects. Finally, using the bundle adjustment method tailored to the special case, i.e. static camera and constant rotation angle, the 3D structure of the object is recovered and the camera parameters are further refined simultaneously.

Different from the existing self-calibration methods as reviewed in Section 2.1.2, our algorithm makes use of constant inter-frame camera motion, i.e. a 3D rigid displacement described by the relative orientation and translation of two cameras. Inter-frame essential matrices are invariant in this case, since the essential matrix depends only on relative camera motion. In addition, we develop a novel linear algorithm for estimating the relative focal lengths of a camera for different frames. The input of the algorithm is only a set of fundamental matrices, and therefore there is no need for projective bundle adjustment before self-calibration.

The rest of the chapter is organized as follows. In Section 5.2, a practical calibration method, making use of constant inter-frame motion, is developed. A simple linear solution is also given which can be used as an initialization. Section 5.3 provides a two-stage optimization method. The

method is then validated through the experiments on both computer simulation and real data in Section 5.4. Finally, Section 5.5 concludes the chapter with discussions on this work.

5.2 The Method

It is well known that when the projective image measurements alone are used it is only possible to recover the scene up to an unknown projective transformation [48, 65]. Additional scene, motion or calibration constraints are required for a metric or Euclidean reconstruction. We also use the constraints on camera internal parameters similar to previous self-calibration methods. However, the main difference is that constant inter-frame motion is exploited in this work.

5.2.1 Self-calibration using Constant Inter-frame Motion

In this section, we first elaborate on the equality between the scenario where the camera is static and the object is rotating around an unknown axis, and the case where the object is fixed while the camera is both rotating and translating.

The i^{th} camera projection matrix can be factorized as $\mathbf{P}_i = \mathbf{K}_i[\mathbf{R} \mid \mathbf{t}]$, since our camera is static and thus has the same $\mathbf{R}$ and $\mathbf{t}$ through all views. We are interested in the case where the relative rotation angle between views i and $i+1$ are constant (Figure 5.1). Let $\mathbf{R}_\theta$ denotes the 3×3 orthonormal rotation matrix of the object, which has only one degree of freedom from

θ. Therefore, after applying the rotation, the projective transformation of the i^{th} frame becomes $\mathbf{K}_i\ [\mathbf{R}\mathbf{R}_\theta^i \mid \mathbf{t}]$.This means that the new camera center is located at $-(\mathbf{R}\mathbf{R}_\theta^i)^\mathbf{T}\mathbf{t}$, with new rotation matrix $\mathbf{R}\mathbf{R}_\theta^i$. Note that the equality is also true for non constant rotations.

Then let us rewrite the i^{th} camera matrix such that the world origin coincides with the i^{th} camera center,

$$\mathbf{x}_i \sim \mathbf{K}_i\ [\mathbf{R}\mathbf{R}_\theta^i \mid \mathbf{t}]\ \mathbf{X} = \mathbf{K}_i\ [\mathbf{I} \mid \mathbf{0}] \begin{bmatrix} \mathbf{R}\mathbf{R}_\theta^i & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}.$$

For the $(i+1)^{th}$ view, we can derive that:

$$\mathbf{x}_{i+1} \sim \mathbf{K}_{i+1}[\mathbf{R}\mathbf{R}_\theta\mathbf{R}^T|(\mathbf{I} - \mathbf{R}\mathbf{R}_\theta\mathbf{R}^T)\mathbf{t}] \begin{bmatrix} \mathbf{R}\mathbf{R}_\theta^i & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}.$$

Therefore we obtain the essential matrix as

$$\mathbf{E}_{i,i+1} = [(\mathbf{I} - \mathbf{R}\mathbf{R}_\theta\mathbf{R}^T)\mathbf{t}]_\times \mathbf{R}\mathbf{R}_\theta\mathbf{R}^T, \tag{5.1}$$

where $[\cdot]_\times$ is the notation for the skew symmetric matrix characterizing the cross product. Since $\mathbf{R}$, $\mathbf{t}$ and $\mathbf{R}_\theta$ are all constants, the inter-frame essential matrices are invariant.

It is possible to use the invariance property of the inter-frame essential matrices to solve for the camera matrices, $\mathbf{K}_i$, given the set of fundamental matrices that encapsulate the intrinsic projective geometry between two views. The equality of essential matrices can be expressed as:

$$\mathbf{K}_{i+2}^T\mathbf{F}_{i+1,i+2}\mathbf{K}_{i+1} \sim \mathbf{K}_{i+1}^T\mathbf{F}_{i,i+1}\mathbf{K}_i, \tag{5.2}$$

where $\mathbf{F}_{i,i+1}$ is the fundamental matrix between the i^{th} and $(i+1)^{th}$ views. A solution may be obtained using a non-linear least squares algorithm. The parameters to be computed are the

unknown intrinsic parameters of each calibration matrix $\mathbf{K}_i$ and the following criterion should be minimized:

$$\min \sum_{i=1}^{n-2} \|\mathbf{K}_{i+2}^T \mathbf{F}_{i+1,i+2} \mathbf{K}_{i+1} - \mathbf{K}_{i+1}^T \mathbf{F}_{i,i+1} \mathbf{K}_i\|_F^2, \tag{5.3}$$

where the subscript F indicates the use of the Frobenius norm, and $\mathbf{K}_{i+2}^T \mathbf{F}_{i+1,i+2} \mathbf{K}_{i+1}$ and $\mathbf{K}_{i+1}^T \mathbf{F}_{i,i+1} \mathbf{K}_i$ are both normalized to have unit Frobenius norm. It is also important to enforce that two of the essential matrices' singular values are equal and the third one is zero. In our implementation, we found that the final results are sensitive to errors in the computed fundamental matrices. Therefore, we recommend the methods [138, 199] that minimize the reprojection errors to compute the fundamental matrices between pairs of images.

5.2.2 Linear approach

To obtain an initial starting point, we propose a linear approach to compute an approximate solution for the calibration. This linear solution can be obtained by assuming zero skew, known aspect ratio and the principal point. For instance, we set the principal point $\mathbf{u}_0$ to $(0, 0)$, and the aspect ratio to one. These assumptions yield:

$$\mathbf{K}_{i+1}^T \mathbf{F}_{i,i+1} \mathbf{K}_i = \begin{bmatrix} f_{i+1} F_i^1 f_i & f_{i+1} F_i^2 f_i & f_{i+1} F_i^3 \\ f_{i+1} F_i^4 f_i & f_{i+1} F_i^5 f_i & f_{i+1} F_i^6 \\ f_i F_i^7 & f_i F_i^8 & F_i^9 \end{bmatrix}, \tag{5.4}$$

where f_i and f_{i+1} are focal lengths of i^{th} and $(i+1)^{th}$ cameras separately, and F_i^k denotes, in a row-major order vector, the components of $\mathbf{F}_{i,i+1}$. From the equation (5.4), and the equivalence

property of the essential matrix, one obtains

$$
\begin{aligned}
\lambda_{i-1,i} f_i F_{i-1}^1 f_{i-1} &= \lambda_{i,i+1} f_{i+1} F_i^1 f_i, \\
\lambda_{i-1,i} f_i F_{i-1}^2 f_{i-1} &= \lambda_{i,i+1} f_{i+1} F_i^2 f_i, \\
\lambda_{i-1,i} f_i F_{i-1}^3 &= \lambda_{i,i+1} f_{i+1} F_i^3, \\
\lambda_{i-1,i} f_i F_{i-1}^4 f_{i-1} &= \lambda_{i,i+1} f_{i+1} F_i^4 f_i, \\
\lambda_{i-1,i} f_i F_{i-1}^5 f_{i-1} &= \lambda_{i,i+1} f_{i+1} F_i^5 f_i, \\
\lambda_{i-1,i} f_i F_{i-1}^6 &= \lambda_{i,i+1} f_{i+1} F_i^6, \\
\lambda_{i-1,i} f_{i-1} F_{i-1}^7 &= \lambda_{i,i+1} f_i F_i^7, \\
\lambda_{i-1,i} f_{i-1} F_{i-1}^8 &= \lambda_{i,i+1} f_i F_i^8, \\
\lambda_{i-1,i} F_{i-1}^9 &= \lambda_{i,i+1} F_i^9,
\end{aligned}
\tag{5.5}
$$

where $\lambda_{i-1,i}, \lambda_{i,i+1} \in \mathbb{R}$. In the cases where the elements F_i^9 of the fundamental matrices are not zero, the focal lengths, f_{i-1}, f_i and f_{i+1}, can be obtained from equations in (5.5) by the left null space of the following matrix:

$$
\begin{bmatrix}
F_i^9 F_{i-1}^1 & F_i^9 F_{i-1}^2 & 0 & F_i^9 F_{i-1}^4 & F_i^9 F_{i-1}^5 & 0 & F_i^9 F_{i-1}^7 & F_i^9 F_{i-1}^8 \\
0 & 0 & F_i^9 F_{i-1}^3 & 0 & 0 & F_i^9 F_{i-1}^6 & -F_{i-1}^9 F_i^7 & -F_{i-1}^9 F_i^8 \\
-F_{i-1}^9 F_i^1 & -F_{i-1}^9 F_i^2 & -F_{i-1}^9 F_i^3 & -F_{i-1}^9 F_i^4 & -F_{i-1}^9 F_i^5 & -F_{i-1}^9 F_i^6 & 0 & 0
\end{bmatrix}. \tag{5.6}
$$

When more images are available, the linear estimation of the focal lengths $(f_i)_{i=1}^n$ can be given by the null space of the $\mathcal{A}_{8(n-2)\times n}$, where

$$\begin{aligned}
\mathcal{A}_{8i-7} &= [\mathbf{0}^T_{(i-1)\times 1},\ F^9_{i+1}F^1_i,\ 0,\ -F^9_iF^1_{i+1},\ \mathbf{0}^T_{(n-2-i)\times 1}],\\
\mathcal{A}_{8i-6} &= [\mathbf{0}^T_{(i-1)\times 1},\ F^9_{i+1}F^2_i,\ 0,\ -F^9_iF^2_{i+1},\ \mathbf{0}^T_{(n-2-i)\times 1}],\\
\mathcal{A}_{8i-5} &= [\mathbf{0}^T_{i\times 1},\ F^9_{i+1}F^3_i,\ -F^9_iF^3_{i+1},\ \mathbf{0}^T_{(n-2-i)\times 1}],\\
\mathcal{A}_{8i-4} &= [\mathbf{0}^T_{(i-1)\times 1},\ F^9_{i+1}F^4_i,\ 0,\ -F^9_iF^4_{i+1},\ \mathbf{0}^T_{(n-2-i)\times 1}],\\
\mathcal{A}_{8i-3} &= [\mathbf{0}^T_{(i-1)\times 1},\ F^9_{i+1}F^5_i,\ 0,\ -F^9_iF^5_{i+1},\ \mathbf{0}^T_{(n-2-i)\times 1}],\\
\mathcal{A}_{8i-2} &= [\mathbf{0}^T_{i\times 1},\ F^9_{i+1}F^6_i,\ -F^9_iF^6_{i+1},\ \mathbf{0}^T_{(n-2-i)\times 1}],\\
\mathcal{A}_{8i-1} &= [\mathbf{0}^T_{(i-1)\times 1},\ F^9_{i+1}F^7_i,\ -F^9_iF^7_{i+1},\ \mathbf{0}^T_{(n-1-i)\times 1}],\\
\mathcal{A}_{8i} &= [\mathbf{0}^T_{(i-1)\times 1},\ F^9_{i+1}F^8_i,\ -F^9_iF^8_{i+1},\ \mathbf{0}^T_{(n-1-i)\times 1}],
\end{aligned} \tag{5.7}$$

here $\mathcal{A}_j$ denotes the j^{th} row of the matrix $\mathcal{A}_{8(n-2)\times n}$.

From the null space of $\mathcal{A}_{8(n-2)\times n}$, we have a solution for the estimation of the focal lengths up to a global scale κ. There are several options to compute κ. One possibility is to pick κ that best enforces the Huang-Faugeras constraint of equality of singular values of the Essential matrices [16, 110] for non-critical motion sequences. In our implementation, we compute the ratios ρ_i of f_i over f_1 (f_1 could be the focal length of any reference image which is without loss of generality assumed to be the first view), and thus compensate the effects of varying focal lengths for each image point, $\mathbf{x}_{ij}$ of the i^{th} image, as $\hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij}/\rho_i$. We then use the existing method [81] to obtain an initial solution of the focal length, f_1, of the first image. Other focal lengths, f_i, can be simply computed as $f_i = \rho_i f_1$.

However, the above method using equation (5.7) will fail when the element F_i^9 of the fundamental matrices are zeros. Note that this special case is easy to be detected since all the essential matrices are equal up to an unknown scale. In other words, we only need to check the element, F_i^9, of one inter-frame fundamental matrix $\mathbf{F}$. It is shown by [110] that whenever the optical axes of the i^{th} and $(i+1)^{th}$ cameras intersect, F_i^9 is equal to zero, since, in this case, the principal points must satisfy the epipolar constraint, i.e. ${\mathbf{u}_0^{i+1}}^T \mathbf{F}_{i,i+1} \mathbf{u}_0^i = 0$, where $\mathbf{u}_0^{i+1} = \mathbf{u}_0^i = [0\ 0\ 1]^T$. While the case where the optical axes of the two cameras intersect is a critical motion for Kruppa-based methods [93], the constant inter-frame motion is still able to provide enough constraints for the computation of the relative focal lengths. For example, one can check the following equations based on the equivalence of the inter-frame essential matrices

$$F_i^n F_{i+1}^m f_{i+1} - F_i^m F_{i+1}^n f_i = 0 \qquad m = 1, 2, 4, 5; \quad n = 3, 6 \tag{5.8}$$

$$F_i^n F_{i+1}^m f_{i+2} - F_i^m F_{i+1}^n f_{i+1} = 0 \qquad m = 1, 2, 4, 5; \quad n = 7, 8. \tag{5.9}$$

Similar to $\mathcal{A}_{8(n-2)\times n}$, we can build another matrix $\mathcal{A}'_{16(n-2)\times n}$, whose null space provides the solution of the focal lengths up to a scale κ. In this case, we still can use existing method [81] to obtain κ, although the Huang-Faugeras constraint will fail.

5.3 Two-stage Optimization

Once the initial approximation for the focal lengths of the cameras has been found by the linear algorithm, a full nonlinear optimization can be run. The nonlinear optimization can be divided into two stages that gradually refine the initial solution obtained from the previous subsections.

5.3.1 Conic enforcement after compensation

Similar to [81], we first improve the results by enforcing the global constraint that the projected trajectories of 3D points should be conics after compensating for the focusing and zooming effects. The main advantage of enforcing conic constraint is that it is intrinsically a multiple view approach as all geometric information from the whole sequence is nicely summarized in the conics as argued by [81]. Practically, conic enforcement efficiently improves the results as shown in Section 5.4.1.

The image points $x_{i,j}$ can be compensated as,

$$\hat{\mathbf{x}}_{ij} = \mathbf{K}_1 \mathbf{K}_i^{-1} \mathbf{x}_{ij}, \tag{5.10}$$

where $\mathbf{K}_1$ is the camera calibration matrix of a reference view, which is without loss of generality assumed to be the first view. After the compensation, the conic property of the correspondence tracks are fully restored, where the entities related to the conic and plane motion become fixed again, such as the rotation axis $\mathbf{l}_s$, horizon line $\mathbf{l}_\infty$, and circular points, $\mathbf{i}$ and $\mathbf{j}$, shown in Figure 5.3.

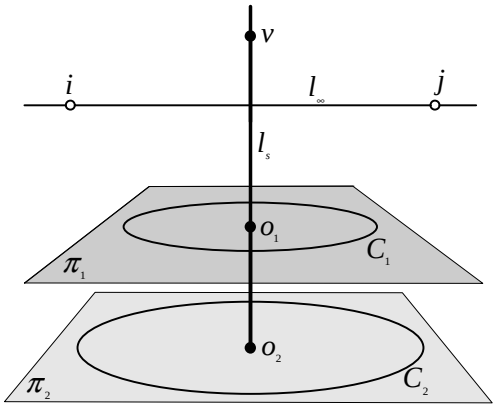


Figure 5.3: The entities related to the geometry of a single axis motion observed by a fixed camera. The fixed entities include the rotation axis, $\mathbf{l}_s$, the horizon line $\mathbf{l}_\infty$, and the circular points, **i** and **j**, the vanishing point, $\mathbf{v}$ of the rotation axis. The projection, $\mathbf{o}_i$, of the center of one circle $\mathbf{C}_i$ is the pole of the horizon line $\mathbf{l}_\infty$ with respect to conic $\mathbf{C}_i$ as $\mathbf{o}_i = \mathbf{C}_i^{-1}\mathbf{l}_\infty$.

Given a conic $\mathbf{C}_j$, there is a homography $\mathbf{H}_j$ to map $\mathbf{C}_j$ into an unit circle, $\mathbf{O}$, such that

$$\mathbf{O} = \mathbf{H}_j^{-T}\mathbf{C}_j\mathbf{H}_j^{-1},$$

where $\mathbf{H}_j$ can be parameterized as [104, 81]

$$\mathbf{H}_j = \begin{bmatrix} s_j & 0 & -\mu_j \\ 0 & s_j & -\nu \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\beta} & -\frac{\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix}, \tag{5.11}$$

where s_j is the scale that enforces the radius of the circle $\mathbf{O}$ to be one, $(\mu_j, \nu, 1)$ is the pole, o_j (Figure 5.3), of $\mathbf{l}_\infty$ with respect to the conic $\mathbf{C}_j$, $(\alpha \pm i\beta, 1, 0)^T$ are the circular points **i** and **j**, and $(l_1, l_2, l_3)^T$ is the vanishing line $\mathbf{l}_\infty$. Basically, the parameters l_1, l_2, l_3, α and β are fixed, since circular points and vanishing line are fixed entities. In addition, ν can be assumed to be constant

in that the pole is constrained by the fixed rotation axis $\mathbf{l}_s$. Given m 3D points, therefore, a total of $6 + 2m$ parameters needs to be estimated by minimizing the following MLE function:

$$\arg\min_{\Theta_1} \sum_{i=1}^{n} \sum_{j=1}^{m} d^2(\hat{\mathbf{x}}_{ij}, \mathbf{C}_j), \tag{5.12}$$

where $\Theta_1 = \{l_1, l_2, l_3, \alpha, \beta, \nu, s_j, \mu_j\}$, $d^2(\hat{\mathbf{x}}_{ij}, \mathbf{C}_j)$ are distance function from point, $\hat{\mathbf{x}}_{ij}$, to conic $\mathbf{C}_j$, defined as

$$d^2(\hat{\mathbf{x}}_{ij}, \mathbf{C}_j) = \begin{cases} \frac{(\hat{\mathbf{x}}_{ij}\mathbf{C}_j\hat{\mathbf{x}}_{ij})^2}{4((\mathbf{C}_j\hat{\mathbf{x}}_{ij})_1^2 + (\mathbf{C}_j\hat{\mathbf{x}}_{ij})_2^2)} & \text{if } (\hat{\mathbf{x}}_{ij} \in \mathbf{C}_j) \\ 0 & \text{otherwise} \end{cases} \tag{5.13}$$

where $(\mathbf{C}_j\hat{\mathbf{x}}_{ij})_i$ is the i-th component of $\mathbf{C}_j\hat{\mathbf{x}}_{ij}$. After substituting $\hat{\mathbf{x}}_{ij}$ with Equation (5.10), we obtain

$$\arg\min_{\Theta_1, \mathbf{K}_i} \sum_{i=1}^{n} \sum_{j=1}^{m} d^2(\mathbf{K}_1\mathbf{K}_i^{-1}\mathbf{x}_{ij}, \mathbf{C}_j). \tag{5.14}$$

This cost function is minimized using the standard Levenberg- Marquart algorithm [114].

The $6 + 2m$ parameters are initialized as follows. First, the focusing and zooming effects are compensated by using Equation (5.10). Second, each conic is fitted to corresponding points from at least five views. Third, the pole of each conic with respect to the vanishing line is calculated as shown in Figure 5.3, and the point on the rotation axis l_s which is nearest to the pole, o_i, is used to estimate the initial value of μ_i. Fourth, the radius of each 3D circle, transformed from each imaged conic, determines the initial value of s_j. Finally, each conic is mapped to a unit circle with center at the origin and the points on the conic is mapped to the points near the unit circle for the optimal procedure.

5.3.2 Reconstruction using bundle adjustment

After the refined camera matrices are obtained, the 3D points or structure can be determined by triangulation from two or more views [74]. To minimize the overall reconstruction errors and to further refine the estimated camera parameters, here we use a bundle adjustment approach [168] explicitly enforcing another available constraint: static camera and constant rotation angle. Given n images and m corresponding image points, the maximum likelihood estimate (MLE) can be obtained:

$$\arg\min_{\Theta_2} \sum_{i=1}^{n} \sum_{j=1}^{m} d^2(\mathbf{x}_{ij}, \mathbf{K}_i[\mathbf{R}\mathbf{R}_\theta^i|\mathbf{t}]\mathbf{X}_j), \tag{5.15}$$

where $\Theta_2 = \{\mathbf{K}_i, \mathbf{R}, \mathbf{R}_\theta, \mathbf{t}, \mathbf{X}_j\}$, and $d(\cdot,\cdot)$ is the distance function between the image measurement $\mathbf{x}_{ij}$ and the projection of the estimated 3D point $\mathbf{X}_j$. Nevertheless, as shown in [52, 75], the circular motion has the fundamental ambiguity on the vertical apex, v (Figure 5.3), which causes unknown ratios between the horizontal and vertical direction for the 3D reconstruction. To remove this ambiguity, we assume a unit aspect ratio and zero skew for all cameras and specify a reasonable choice of the aspect ratio of the object.

Similar to most other self-calibration methods, such as [134, 4], we also have difficulty to precisely estimate the principal points because the principal point $\mathbf{u}_0$ is known to be a poorly constrained parameter which tends to fit to noise. In practice, we notice that $\mathbf{u}_0$ is mostly located close to the image center. Using this prior information, we model the expectation of the principal point as a Gaussian distribution, which has its mean at the image center $\bar{\mathbf{u}}_0$, with the uncertainties $\Sigma_{\mathbf{u}_0} = diag(\sigma_u^2, \sigma_v^2)$. Therefore, we apply the prior information of principal point on Equation

(5.15). Consequently, the bundle adjustment is rewritten as:

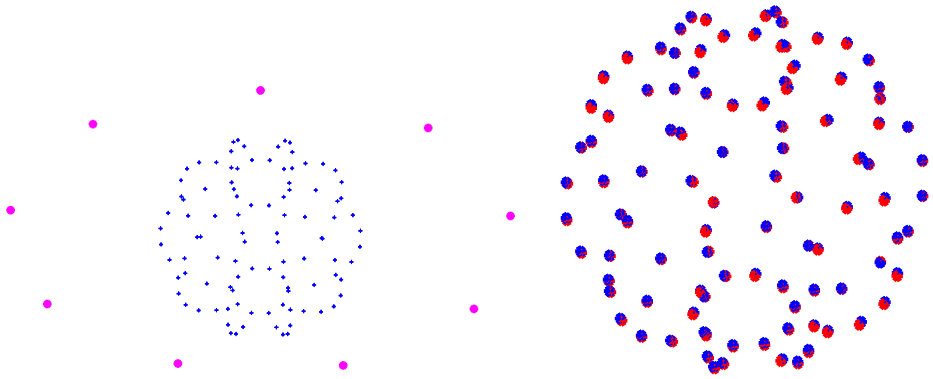
$$\arg\min_{\Theta_2} \sum_{i=1}^{n} \left(\sum_{j=1}^{m} d^2(\mathbf{x}_{ij}, \mathbf{K}_i[\mathbf{R}\mathbf{R}_\theta^i|\mathbf{t}]\mathbf{X}_j) + (\mathbf{u}_0^i - \bar{\mathbf{u}}_0)^T \Sigma_{\mathbf{u}_0}^{-1} (\mathbf{u}_0^i - \bar{\mathbf{u}}_0) \right), \tag{5.16}$$

where $\mathbf{u}_0^i$ is the estimate of the principal point for each view. Without further mention, we use $1/10$ image width and height as σ_u and σ_v, and $1/2$ image width and height as $\bar{u}_0$ and $\bar{v}_0$ respectively.

Note that our optimization process differs from the general reconstruction, e.g. the method in Section 4.2.3, in that it explicitly encodes the specific non-general motion, i.e. constant $\mathbf{R}$, $\mathbf{t}$, and $\mathbf{R}_\theta$. Consequently, a total of $3m + 4$ parameters must be estimated for m views, where three is the number of degrees of freedom of $\mathbf{K}$ (note that we enforce zero skew and unit aspect ratio), and four includes three rotation angles in $\mathbf{R}$ and one constant but unknown angle θ (Figure 5.1) in $\mathbf{R}_\theta$. This is a considerable saving over the $9m$ that would be required for a projective reconstruction of a general motion sequence if we make the same assumptions on the camera internal parameters, which reduce the number of degrees of freedom of a projection matrix $\mathbf{P}$ of a pinhole camera from eleven to nine.

5.4 Experimental Results

The proposed approach has been tested on both simulated and real image sequences. First, a synthetic image sequence is used to assess the quality of the algorithm under simulated circumstances. Both the amount of noise on the projected image points and on the rotation angles of the objects



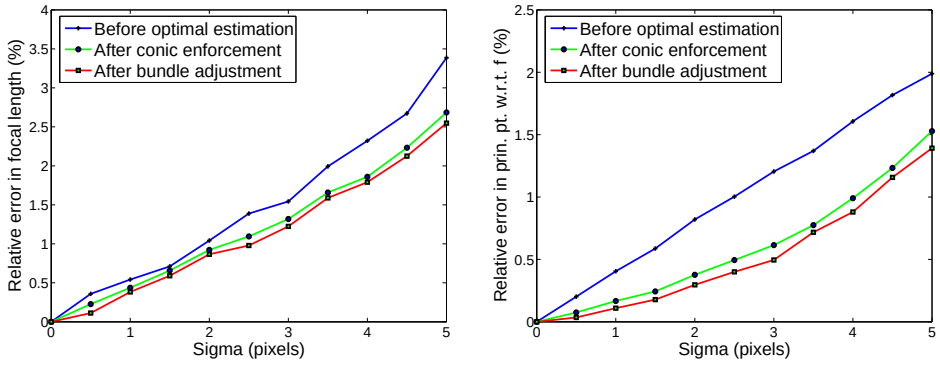
(a) (b)

Figure 5.4: (a) A view of geometrically equivalent sequences used for simulation, where magenta points denote the positions of cameras. (b) Reconstructed 3D points, where blue cubes denote the ground truth while red cubes are reconstructed ones at the noise level $\sigma = 2.5$ pixels.

are varied. Then results are given for real image sequences to demonstrate the usability of this proposed solution.

5.4.1 Computer simulation

The simulations are carried out on a sequence of views of a synthetic scene, which consists of 100 points uniformly distributed on a sphere with a radius of 200 units and centered at the origin. Our synthetic camera is located in front of the scene at a distance of 500 units with three rotation angles (20°, 20° and 15°) between the world coordinate system and the camera coordinate system. In addition to a unit aspect ratio and zero skew, the camera's other internal parameters are chosen

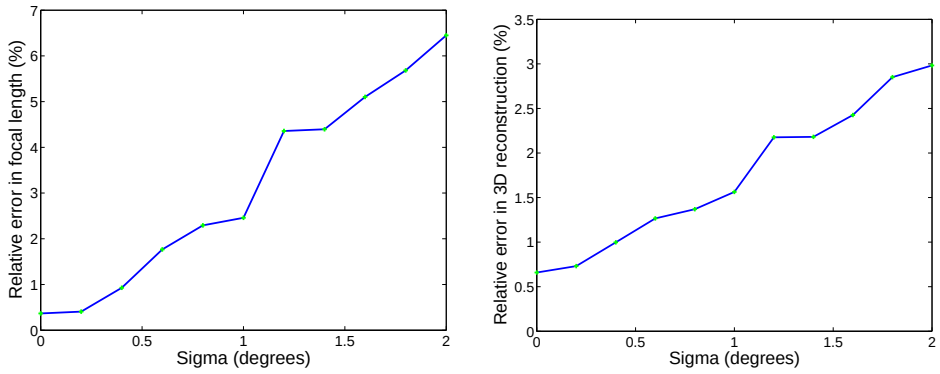


(a) (b)

Figure 5.5: Performance of the focal length, f, and the principal point, $\mathbf{u}_0$, in a function of noise levels: (a) relative error of f, and (b) relative distance of the principal point $\mathbf{u}_0$ with respect to the true focal lengths.

as follows. The focal lengths are different for each view, randomly chosen with an expected value of 1000 (in pixels) and a standard deviation of 250. To avoid the case that the chosen focal lengths fall outside the reasonable range, e.g. below zero, we limit them to vary between 750 and 1250. The principal point, $\mathbf{u}_0$, had an expected value of $[0\ 0]^T$ with a standard deviation of $20\sqrt{2}$. An example view of the equivalent scene, where the camera is moving and the object is stationary, is shown in Figure 5.4 (a).

Performance Versus Pixel Error: To assess the performance versus noise on the projected image points, nine views are generated to compute the camera matrices. Gaussian noise with zero mean and a standard deviation of $\sigma \leq 5.0$ pixels was added to the projected image points. The estimated camera parameters were then compared with the ground truth. As argued by [173] and [202], the relative difference with respect to the focal length rather than the absolute error is a



(a) (b)

Figure 5.6: Performance of focal length and 3D metric reconstruction in a function of rotation angle errors: (a) relative error of focal length and (b) relative 3D metric error. All results shown here are averaged over 100 independent trials.

geometrically meaningful error measure of camera internal parameters. Therefore, we measured the relative error of focal length, f, and the principal point, $\mathbf{u}_0$, while varying the noise level from 0.5 pixels to 5.0 pixels. At each noise level, we perform 100 independent trials, and the averaged results of the proposed self-calibration algorithm are shown in Figure 5.5. Errors increase almost linearly with respect to the noise level for both focal lengths and principal points. Using our two-stage non-linear optimization, the results are refined from a coarse starting point to a fine level for both f and $\mathbf{u}_0$. After the first stage conic enforcement, it reduces on average around 18.5% (range from 7.4% to 36.7%) errors of the estimated focal lengths. Then these errors are further reduced by another average 11% at the second stage bundle adjustment after enforcing the constant rotation angle. In our experiment, we even increase the σ up to 5 pixels. For $\sigma = 2.5$, a typical large noise in the practical calibration, the relative error of focal length f is 1.0%. Figure 5.4 (b) shows the

3D reconstructed scene in one trial. The maximum relative error of f (resp. $\mathbf{u}_0$) is around 2.55% (resp. 1.39%) when $\sigma \leq 5.0$.

Performance Versus Rotation Angle Error: Another experiment (nine views) is carried out to evaluate how sensitive the algorithm is to noise in the rotation angles. Gaussian noise with zero mean and a standard deviation of $\sigma \leq 2.0$ degrees was added to the rotation angles. Considering the fact that extracted feature points will in practice be affected by noise, we also added a typical noise level of $\sigma = 1.0$ pixels to all projected image points. The final results after optimal estimation are shown in Figure 5.6. The influence of the orientation noise is larger than that of pixel noise (see Figure 5.5), which of course depends on the absolute rotation angle between the views. Note that this coincides with the observation by Frahm and Koch [54], in which case this is more evident since they have smaller rotation angles. The errors in both focal lengths and 3D reconstruction increase almost linearly with respect to the rotation angle noises. Notice also that the errors do not go to zero as noise goes towards zero due to the added noise in image projections.

5.4.2 Real data

The first real sequence is the Tylenol sequence from Columbia Object Image Library (COIL-20). The COIL sequences have previously resisted structure from motion extraction, due to their low feature counts and variable focal length, which this work provides the machinery to overcome. We use 18 frames out of the original 72 views of the box as shown in Figure 5.7. The tracks of the

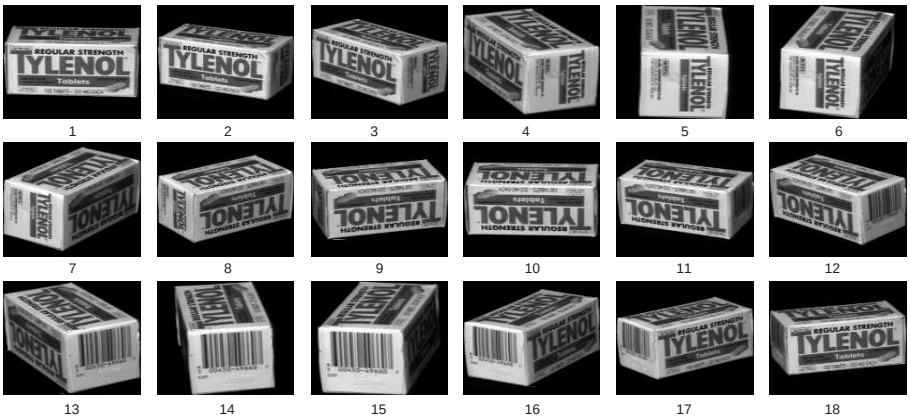
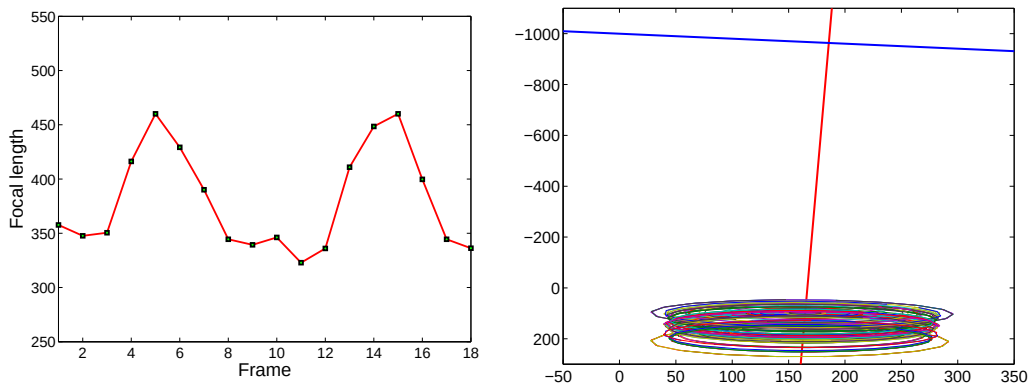


Figure 5.7: Eighteen views of the Tylenol sequence.

corresponding points estimated using our previous work [195] are shown in Figure 5.9 (a), and the final determined focal lengths for these images are shown in Figure 5.8 (a). The estimated focal lengthes are consistent with the real sequence. For example, the camera zoomed in considerably to capture the 5^{th} and 15^{th} frames while it zoomed out when shooting the 10^{th} frame. To evaluate the proposed method, we first compensate the frames according to the final estimated calibration matrices by using the 7^{th} frame as the reference. The fitted conics and estimated rotation axis are shown in Figure 5.9 (b-d) for three compensated frames (frames 7, 5, and 11). We also show the conics, rotation axis and horizontal line of the compensated frames in Figure 5.8 (b). Finally, piecewise planar model with mapped texture is shown in Figure 5.10.

We also tested our approach on the popular dinosaur sequence from the University of Hannover. The sequence contains 36 views of a dinosaur located on a turn-table which is rotating with a constant angular motion of 10 degrees per frame. One frame with tracked points is shown in

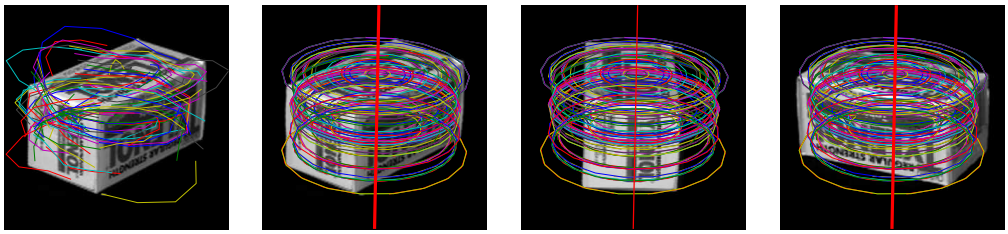


(a) (b)

Figure 5.8: (a) Computed focal lengths of the Tylenol sequence. (b) The conics, rotation axis and horizontal line of the compensated frames. Note: we scale the vertical direction to show the all entities.

Figure 5.11 (a). The computed focal length for the image sequence is shown in Figure 5.12 (a). The results are consistent to the known truth that the focal lengths are fixed.

In another dinosaur sequence, the focal lengths of the camera is set to change in a zigzag fashion ($0.8 - 1.0 - 1.2$), by rescaling the original images. Three consecutive frames are shown in Figure 5.13. When the static camera is free to zoom and focus, the 3D circular trajectory is not projected to a conic anymore (Figure 5.11 (b)). The computed focal lengths for the dinosaur sequence is shown in Figure 5.12 (b), which is close to the changing pattern in a zigzag fashion. To estimate the correctness of our proposed method, the visual hull of the dinosaur can be computed [124] as shown in Figure 5.14. The processing of the volume is performed using a resolution in space of 200^3 unit cubes for the bounding box of the dinosaur. Although the error in the estimated visual hull is expected to be higher than a calibrated sequence or a sequence captured by a camera



(a) (b) (c) (d)

Figure 5.9: (a) One original frame (frame 7) of the Tylenol sequence and a subset of the point tracks - only 94 tracks which survived for longer than 4 successive views out of the original 18 views are shown. (b-d) show the estimated conics on compensated frames 7, 5 and 11. The red vertical line is the rotation axis.

with fixed internal parameters, our method provides a powerful solution for reconstruction from circular motion in the presence of changing internal parameters.

5.5 Conclusion

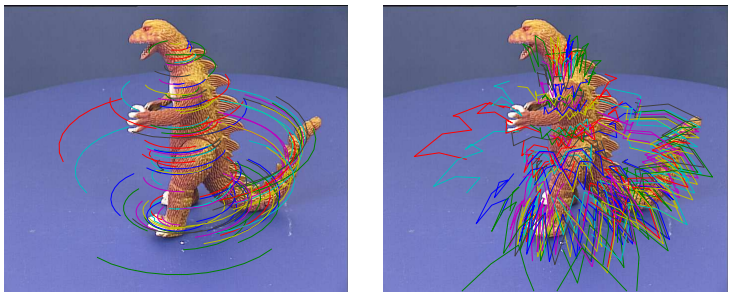
This chapter focused on the problem of self-calibration from an image sequence of an object rotating around a single axis (or equivalently a camera moving on a circle around an object in the scene) in the presence of varying camera internal parameters. Using the invariance property of the inter-frame essential matrices when the rotation angle is constant, we present a new and elegant solution for camera calibration. Compared to the existing methods, we effectively utilize the prior information, such as constant rotation angle and circular motion, develop linear algorithm to find an initial solution assuming zero skew and known aspect ratio and the principal point, and design

(a) (b) (c)

Figure 5.10: Three snapshots of the piecewise planar models with mapped texture of the Tylenol box.

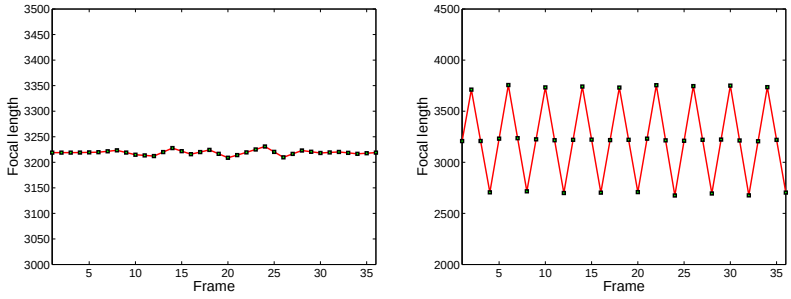
a two-stage optimization approach to gradually refine the camera parameters from coarse to fine. The experimental results demonstrate the usability of this proposed solution.

There are two potential applications of this method. First, it can be used to recover the 3D model of an object, which is useful in model based video post-production applications. Due to the fact that the camera is free to zoom and focus, it can potentially provide higher resolution 3D models. Second, this method can be used for simultaneous calibration of a camera network in surveillance system. One solution to occlusion problem in many heavily crowed surveillance sites (for e.g. airports, subway stations and railway stations) is to use multiple sensors (cameras) [109]. In the general case, e.g. the cameras have the same FOV and the scene geometry is general, the area covered by sensors is maximized by putting them on a circle with equal angular separation, which is geometrically equal to the case of the turn-table sequence as shown in Section 5.2.1.



(a) (b)

Figure 5.11: One frame of the dinosaur sequence and a subset of the point tracks - only 107 tracks which survived for longer than 8 successive views are shown. (a) In the case of a static camera with fixed internal parameters, the point tracks are ellipses which are images of circles. (b) When the static camera is free to zoom and focus, the 3D circular trajectories are not projected to conics anymore.

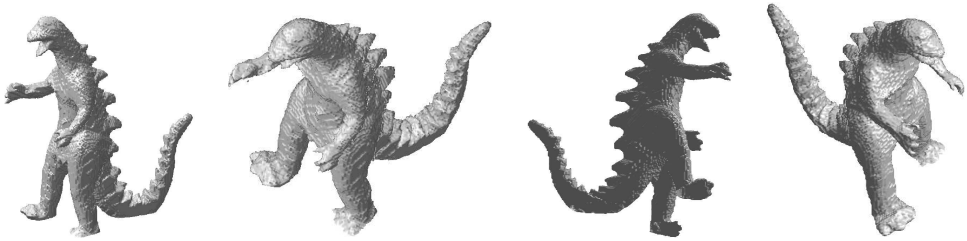


(a) (b)

Figure 5.12: (a) Computed focal lengths of the dinosaur image sequence with fixed focal length, and (b) Computed focal lengths of the dinosaur image sequence with focal length changing in a zigzag fashion.

(a) (b) (c)

Figure 5.13: Three consecutive sample frames of the zigzag dinosaur sequence. These frames are rescaled from the original images by scales 0.8 (a), 1.0 (b) and 1.2 (c).



(a) (b) (c) (d)

Figure 5.14: Four views of the 3D reconstruction of dinosaur from silhouettes.

CHAPTER 6

VIDEO ANALYSIS

In video post-production applications that will be discussed in Chapter 7, camera motion analysis and alignment are important to ensure the geometric correctness and temporal consistency [161]. The above two chapters discussed the accurate 3D methods for recovering the camera calibration and geometry for general and practical video scenes. The recovered camera motion can be directly used in 3D alignment as shown in Section 6.2. However, we are also willing to trade some generality in estimating and aligning camera motion for reduced computational complexity and increased image-based nature.

Particularly, this chapter addresses the problem of synchronizing video sequences of distinct scenes captured by cameras undergoing similar motions. For the general motion and 3D scene, the camera ego-motions are featured by parameters obtained from the fundamental matrices. In the case of special motion, e.g. pure translation or pure rotation, relative translational magnitude or rotation angles are used as camera motion features. These extracted features are invariant to the camera internal parameters, and therefore can be computed without recovering camera trajectories along the image sequences. Consequently, the alignment problem reduces to matching sets of

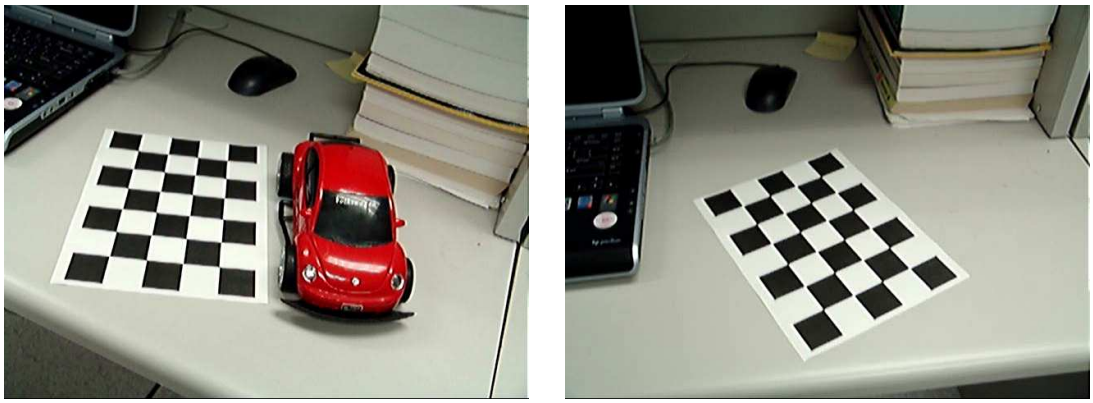
feature points, obtained without any knowledge of other sequences. Experimental results show that our method can accurately synchronize sequences even when they have dynamic timeline maps, and the scenes are totally different and have dense depths.

6.1 Introduction

The problem of synchronizing video sequences has become an active area in computer vision community as described in Section 2.2.3. The proposed approach herein tackles a general problem, i.e. the synchronization of N video sequences of distinct general scenes captured by cameras undergoing similar ego-motions. In this work, two camera motions are defined to be similar when the camera locations and poses in all corresponding time slots are related by a common 3D similarity transformation. In the case of general camera motion, similar camera displacements result in the same relative inter-frame orientation and scaled relative inter-frame translation, i.e. the same essential matrices up to an unknown scale, between the synchronized frame pairs. Our solution is based on the key observation that the equality of the essential matrix is reflected in the uncalibrated fundamental matrix in that its upper-left 2×2 elements remain constant up to an unknown scale. Therefore, for each frame, we can obtain a homogeneous four-dimensional feature vector characterizing the camera ego-motion relative to a reference frame. Of course only the ratios of the elements of those feature vectors are relevant. We call these ratios the *fundamental ratios*.

On the other hand, the fundamental ratios would fail to provide camera ego-motion information for non general camera motion, in which case the number of degrees of freedom of the fundamental matrix is less than the seven of general motion, or the epipolar geometry is undefined, e.g. fixed camera centers or planar scenes. To overcome the degeneracies, we analyze the special motions and provide dedicated features to align them. For special motions, we mainly concentrate on three types of camera motions: pure translation, pure rotation and zooming. In the case of pure translational camera motion, we use the relative translational magnitude by slicing the 3D data volume along the epipolar lines. For cameras with fixed locations undergoing pure rotation, we compute the rotation angles directly from the inter-frame planar homography by eigenvalue decomposition. Finally, the zooming factor is used as the camera motion feature for the zooming video sequences.

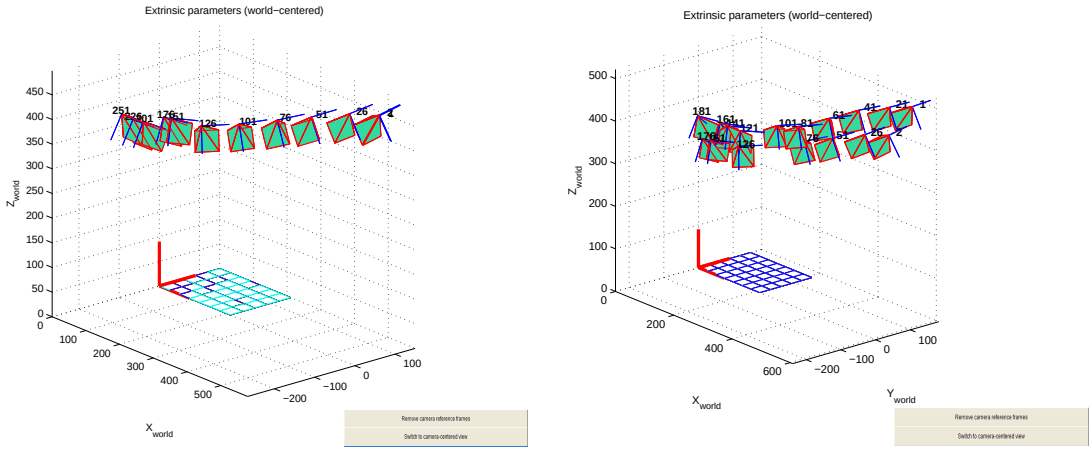
The main advantage of the new method is that it can deal with video sequences of general distinct scenes captured by unknown, either generally or specially, moving cameras. Compared to the traditional synchronization methods which require video cameras observing the same site, our method is able to handle totally distinct scenes. Different from the previous efforts on temporal alignment of non-overlapping sequences, which typically utilizes the inter-sequence relationship and hence inherently involves two sequences, the proposed approach is a 1-sequence process and computes the camera ego-motion for each sequence separately. Consequently, the alignment problem reduces to matching N sets of feature points. Besides its efficiency allowing a combination-free implementation, our algorithm, as a 2D solution, does not involve scene reconstruction or 3D recovery of the camera trajectories [125]. The input of our algorithm is only the point correspondences within each sequence. Finally, the approach is simple to implement.



(a) A source video frame (b) A target video frame

Figure 6.1: Examples frames in source and target video sequences.

This chapter begins with the illustration of how to make use of the explicitly recovered camera motions (See Chapters 4 and 5) for video alignment in Section 6.2. In Section 6.3, the concept of the fundamental ratios for the description of general camera ego-motions is introduced. Section 6.4 then describes a statistical approach to characterize special camera motions such as pan, zoom and track. The motion features for these special camera motions are also provided. Next, in Section 6.5, we present the algorithm and its implemental details. Experimental results on real video sequences of different camera motions are demonstrated in Section 6.6. Finally, we discuss the contributions and provide the future direction in Section 6.7.



(a) source video (b) target video

Figure 6.2: Camera trajectories of both source and target video sequences computed by the method [200].

6.2 3D Alignment Using Recovered Camera Motions

For hand-held shots where camera motions can not be simply related by a similarity transformation, we first estimate the poses of the cameras that captured both the source and the target scenes, and then align these two videos along 3D camera trajectories by minimizing the global differences between the source and the transformed target viewing directions.

The methods described in Chapter 4 and 5 can be used for the camera pose estimation. For example, given two hand-held shots shown in Fig. 6.1, the computed camera positions and poses are shown in Fig. 6.2. For another example, Fig. 6.3 gives the estimated camera trajectories of two hand-held shots in Figure 7.1.

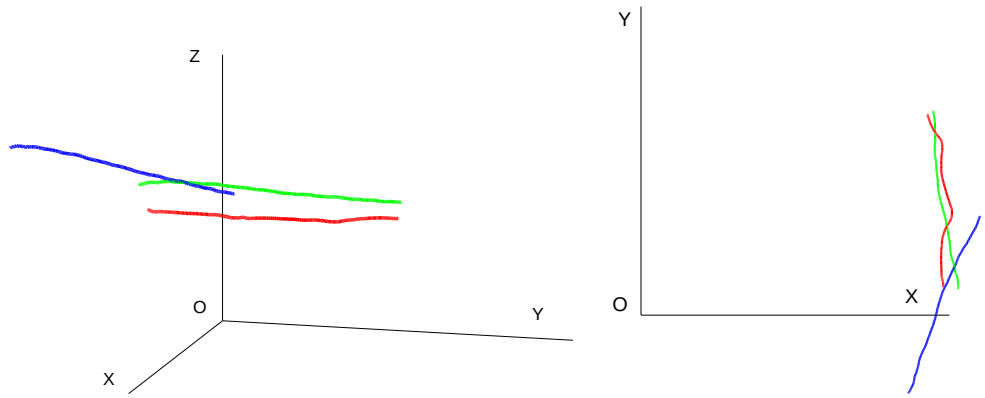
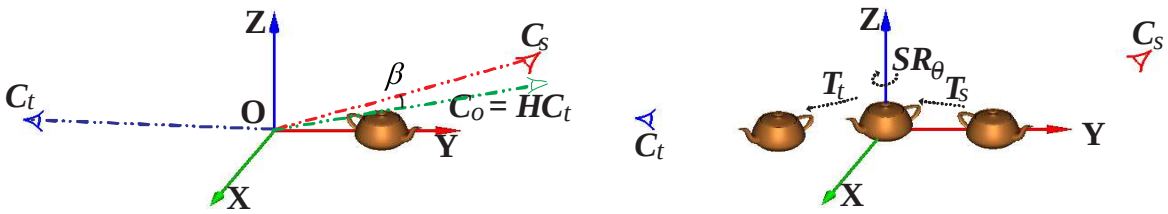


Figure 6.3: 3D camera trajectory alignment between two shots in Figure 7.1. The left is the 3D view and the right side is the top view. The red curve is the source camera trajectory. The blue and green curves are the target camera trajectories before and after alignment.

Theoretically, we have the freedom to choose the world coordinate frames of both the source and target shots, and directly match the source and target frames by minimizing the global differences between the source and the target viewing directions. However, the direct method would not always work well, since the viewing directions of the source ($\mathbf{C}_s$) and target ($\mathbf{C}_t$) cameras may be very different after overlapping their world coordinates as shown in Figure 6.4 (a). Consequently, in the video post-production applications discussed in Chapter 7, if we simply render the object from $\mathbf{C}_t$ using source view captured from $\mathbf{C}_s$, unpleasing distortions may happen as shown in Figure 6.5 (c).

To minimize these distortions, we aim to choose the 3D world coordinate systems for the source and target scenes such that the source and target camera trajectories are globally matched. In other words, we need to find a 4×4 projective transformation matrix $\mathbf{H}$ to transform the specified



(a) (b)

Figure 6.4: (a) Camera trajectory alignment using $\mathbf{H}$. (b) The decomposition of $\mathbf{H}$.

target world coordinate system (Figure 6.5 (b)) to a new one, with the intention that the new target viewing directions, $\mathbf{C}_o = \mathbf{H}\mathbf{C}_t$, are closer to source view directions $\mathbf{C}_s$ (Figure 6.4 (a)), and thus the matted object can be rendered with the least global distortions as shown in Figure 6.5 (d). Generally, the global coordinate transformation matrix $\mathbf{H}$ can be approximately decomposed into four sub-transformations (see Figure 6.4 (b)):

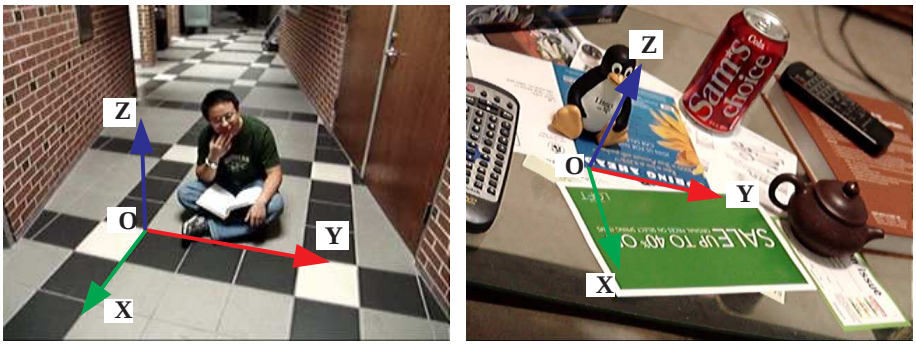
$$\mathbf{H} = \mathbf{T}_t\mathbf{S}\mathbf{R}_\theta\mathbf{T}_s, \tag{6.1}$$

where $\mathbf{T}_s$ is a translation matrix to translate the foreground object to be cut into the 3D origin of the source video, $\mathbf{R}_\theta$ is a rotation matrix around $\mathbf{Z}$ axis, $\mathbf{S}$ is a global scaling matrix, $\mathbf{T}_t$ is the object's destination in the target scene.

As a result, the object is able to move on any reference plane, perpendicular to $\mathbf{Z}$ axis, of the target environment with a flexible scale and rotation. Given camera matrices $\mathbf{P}_s^i$ and $\mathbf{P}_t^j$ for the source frame i and the target frame j, a 3D point, $\mathbf{X}$, inside the object is projected as

$$\mathbf{x}_s^i = \mathbf{P}_s^i\mathbf{X}, \tag{6.2}$$

$$\mathbf{x}_o^j = \mathbf{P}_t^j\mathbf{H}\mathbf{X} = \mathbf{P}_o^j\mathbf{X}, \tag{6.3}$$



(a) Source frame (b) Target frame

(c) Result without alignment (d) Result with alignment

Figure 6.5: Cut a person from (a) and paste it to (b). The world 3D coordinates are superimposed in the source and target frames.

where $\mathbf{x}_o^j$ and $\mathbf{x}_s^i$ are the 2D projections of $\mathbf{X}$, and $\mathbf{P}_o^j$ is the transformed target camera matrix.

To compute $\mathbf{H}$, we enforce the least distortion property that minimizes the viewing orientation differences. Given a camera matrix $\mathbf{P} = [\mathbf{M}\,|\mathbf{t}]$, the 3D camera location is computed as $\mathbf{C} = -\mathbf{M}^{-1}\mathbf{t}$. After translating the 3D world origin to the "Look At" position, which typically locates at the principal point, we can use vector $\vec{\mathbf{C}}$ as the viewing direction of camera $\mathbf{P}$. Therefore, the

view orientation difference between $\mathbf{P}_o^j$ and $\mathbf{P}_s^i$ is computed as

$$\beta_{i,j} = \arccos\left(\frac{\vec{\mathbf{C}}_s^i \cdot \vec{\mathbf{C}}_o^j}{\| \vec{\mathbf{C}}_s^i \| \cdot \| \vec{\mathbf{C}}_o^j \|}\right), \tag{6.4}$$

where $\vec{\mathbf{C}}_s^i$ and $\vec{\mathbf{C}}_o^j$ are the viewing directions of $\mathbf{P}_s^i$, and $\mathbf{P}_o^j$ respectively. Standard nonlinear minimization methods can be utilized now to find the solution for $\mathbf{H}$, given a reasonable initialization, which can be chosen as follows. In practice, we want to transfer the object into the target scene at a specified location, $\mathbf{T}_t$, and a desired scale, $\mathbf{S}$, with an optimal viewing field for the composited video sequence. In addition, we can compute $\mathbf{T}_s$ directly by specifying the object location in the source shot. Finally, the rotation angle θ can be estimated by minimizing the view angles between the ending source and target frames. For example, Figure 6.3 shows the camera trajectories before and after 3D alignment. Once the $\mathbf{H}$ is determined, for each target frame j, the closest source view i can be found by minimizing $\beta_{i,j}$.

6.3 2D Alignment of Similar General Motion

A video sequence $\mathcal{V}$ is an ordered set of images $\{\mathcal{I}_j\}_1^N$ captured by a camera. Assuming a pin-hole camera model and constant camera internal parameters, the camera projection matrix $\mathbf{P}_j$ of the frame $\mathcal{I}_j$ can be factorized as $\mathbf{P}_j = \mathbf{K}[\mathbf{R}_j|\mathbf{t}_j]$, where $\mathbf{K}$ is the simplified camera calibration matrix including the intrinsic parameters, i.e. the focal length f and the principal point $\mathbf{x}_0 = (u_0, v_0)$, and $\mathbf{R}_j$ and $\mathbf{t}_j$ are camera rotation and translation, respectively. As shown in Fig. 6.6, if we have another sequence $\mathcal{V}'$, i.e. $\{\mathcal{I}'_{j'}\}_1^{N'}$, of distinct general scenes captured by a camera undergoing

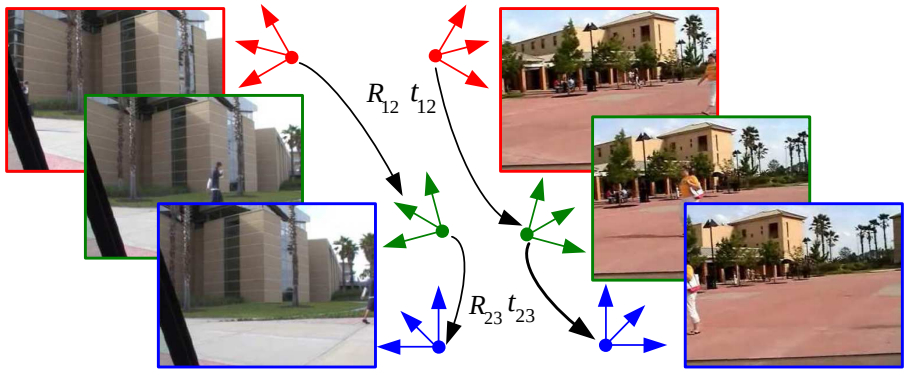


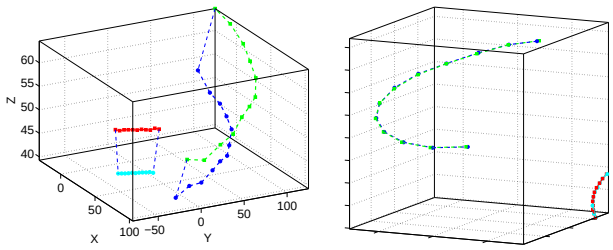
Figure 6.6: Two video sequences $\mathcal{V}$ (left) and $\mathcal{V}'$ (right) of distinct general scenes captured by cameras undergoing the same general movement.

similar motion as that of $\mathcal{V}$, the camera projection matrix $\mathbf{P}'_{j'}$ of frame j' in $\mathcal{V}'$ can be factorized as $\mathbf{P}'_{j'} = \mathbf{K}'[\mathbf{R}_j \mid \mathbf{t}_j]\mathbf{H}_s$, where $j' = c(j)$ is the frame index in $\mathcal{I}'$ corresponding to frame j in sequence $\mathcal{I}$, and the 4×4 similarity transformation matrix $\mathbf{H}_s$ relates the locations and poses of the two synchronized cameras that capture two corresponding frames. The correspondence relationship, $c(\cdot)$, can be dynamic [140] or modeled by a 1D affine model [33, 34, 166].

In the noise-free case, it is simple to verify that the relative translation $\mathbf{t}_{i,j}$ (orientation $\mathbf{R}_{i,j}$) between frames i and j in sequence $\mathcal{V}$ are equal up to an unknown scale (the same) to the relative translation (orientation) between frames $c(i)$ and $c(j)$ in sequence $\mathcal{V}'$. Therefore, we have

$$\mathbf{E}_{i,j} \sim [\mathbf{t}_{i,j}]_\times \mathbf{R}_{i,j} \sim \mathbf{E}'_{c(i),c(j)}, \tag{6.5}$$

where $\sim$ indicates equality up to multiplication by a non-zero scale factor, and $[\cdot]_\times$ is the notation for the skew symmetric matrix characterizing the cross product. The scale ambiguity $\begin{pmatrix} 0.8\mathbf{I} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}$ is demonstrated in Figure 6.7. As a result, when the camera calibration matrices $\mathbf{K}$ and $\mathbf{K}'$ of the sequences $\mathcal{V}$ and $\mathcal{V}'$ are identical, the corresponding uncalibrated fundamental matrices $\mathbf{F}_{i,j}$ and



(a) (b)

Figure 6.7: (a) The locations and poses of the blue and green (respectively the red and cyan) cameras are related by a scale ambiguity, which however are noise-corrupted. (b) Using the fundamental ratio v_g in Equation (6.7), the videos can be temporally aligned. Note that only the first three elements of v_g are plotted in (b).

$\mathbf{F}'_{c(i),c(j)}$ should be equal, which can be used for video synchronization. However, we are interested in a more general case, where $\mathbf{K}$ and $\mathbf{K}'$ are constant but different between the sequences.

In the case of a simplified camera model, i.e. unit aspect ratio and zero skew, it is easy to verify that the upper left 2×2 sub-matrix of $\mathbf{F}$ has the form:

$$\mathbf{F}^{2\times 2} \sim \begin{bmatrix} \epsilon_{1st}\mathbf{t}^s_{i,j}\mathbf{r}^t_1 & \epsilon_{1st}\mathbf{t}^s_{i,j}\mathbf{r}^t_2 \\ \epsilon_{2st}\mathbf{t}^s_{i,j}\mathbf{r}^t_1 & \epsilon_{2st}\mathbf{t}^s_{i,j}\mathbf{r}^t_2 \end{bmatrix}, \tag{6.6}$$

where ϵ_{rst} for $r, s, t = 1, \ldots, 3$ is the permutation tensor, $\mathbf{r}_i$ are columns of the rotation matrix $\mathbf{R}_{i,j}$. The interesting observation of $\mathbf{F}^{2\times 2}$ is that the ratios among its elements F_{ij} are invariant to the camera internal parameters and reflect only the camera ego-motion. In this , we call these ratios the *fundamental ratios*. Therefore, we are able to extract an independent four dimensional feature, $\mathbf{v}_g$, for general camera ego-motion as,

$$\mathbf{v}_g = \text{sign}(F_{11})[F_{11}, F_{12}, F_{21}, F_{22}]/\|\mathbf{F}^{2\times 2}\|_F, \tag{6.7}$$

where $\|\cdot\|_F$ is the Frobenius norm.

It is unlikely for the 4D vector $\mathbf{v}_g$ to uniquely characterize the relative camera ego-motion, which has five degrees of freedom: both the rotation $\mathbf{R}_{i,j}$ and translation $\mathbf{t}_{i,j}$ have three degrees of freedom, but there is an overall scale ambiguity. In addition, there are four possible setups of relative camera position and orientation for the same essential matrix as shown in [75]. However, similar camera ego-motions would result in the same $\mathbf{v}_g$, which can be used to synchronize video sequences as shown in Section 6.5.

6.4 2D Alignment of Similar Special Motion

There are certain special cases of motion, where the number of degrees of freedom of the fundamental matrix is less than the seven of general motion, or the epipolar geometry is undefined. Consequently, the fundamental ratios in Equation (6.7) would fail to provide camera ego-motion information for non general camera motion. In this work, we mainly concentrate on the following three common special motions: pure translation, pure rotation and zoom, and present a statistical method for classifying the special camera motions into the following categories (see Figure 6.8):

1. pure rotation: pan (tilt, swing respectively) with the rotation angle α (β, γ respectively);

2. pure translation: side-way track (boom, forward-outward track respectively) with the translation vector $\Delta\mathbf{t} = (t_x,\ 0,\ 0)^T$ ($\Delta\mathbf{t} = (0,\ t_y,\ 0)^T$, $\Delta\mathbf{t} = (0,\ 0,\ t_z)^T$ respectively),

3. zoom : with the ratio (or zoom factor) s of the camera focal lengths;

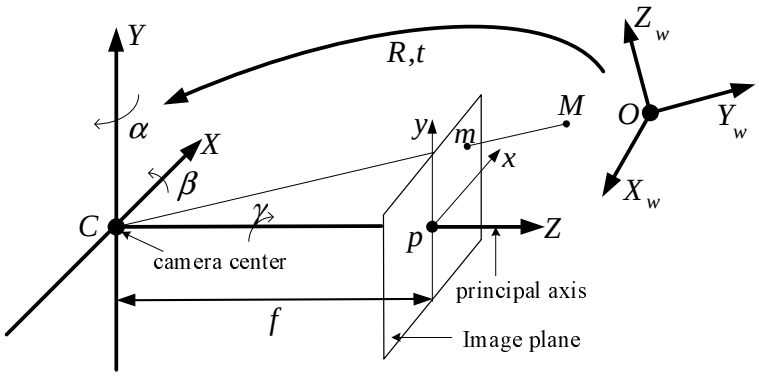


Figure 6.8: The geometry of a pin-hole camera and camera motion parameters. The extrinsic parameters of a pin-hole camera represent the rigid body transformation between the world coordinate system (centered at **O**) and the camera coordinate system (centered at **C**). The intrinsic parameters, e.g. the focal length f, stand for the camera internal geometry.

4. static: with identity inter-frame planar homography;

Except for the third category (zoom), the camera intrinsic parameters in Equation (3.4) are fixed. For each different special camera motion, we give geometric explanation of their properties and the method to compute their motion magnitudes, e.g. the rotation angles and zoom factors, which are used for temporal alignment in Section 6.5.

Existing camera motion classification methods utilizing 2D parametric transformation, as summarized in Section 2.2.1, are reasonable only when the scenarios are restricted to 2D scenes or when the camera position is fixed (static, zooming, pure rotation), while methods [122] using spatio-temporal slices only involve the patterns of different camera motions without geometric explanation or quantification of the camera motion. The proposed method is based on existing works: pattern analysis of image slices in a spatial-temporal volume Ngo [122], Bayesian model estima-

tion [169], and stereo geometry [75]. However, this proposed method provides geometrically meaningful analyzes of the orientations of the patterns depicting motions in slices. In addition, the motion magnitudes, such as translational speed, are numerically quantified. Finally, we consider the perspective effects in slicing the data volume.

In the following discussion, we denote by $\mathbf{x}$ and $\mathbf{x}'$ the images of a 3D scene point $\mathbf{X}$ before and after the camera motion. For simplicity, the world coordinate frame will be chosen to coincide with the camera before moving, so that $\mathbf{x} \sim \mathbf{PX} = \mathbf{K}[\mathbf{I} \mid \mathbf{0}]\mathbf{X}$ (see Section 3.4).

6.4.1 Pure Rotation

Pure rotation is a very important camera motion, e.g. for the PTZ cameras used in video surveillance systems. It is known [75] that the corresponding points $\mathbf{x}$ and $\mathbf{x}'$ before and after the rotation are related by an infinite planar homography $\mathbf{H}_\infty = \mathbf{KRK}^{-1}$, where $\mathbf{R}$ is the relative rotation, which has only one degree of freedom from the rotation angle: α (pan), β (tilt) or γ (zoom).

As a similarity transformation, $\mathbf{H}_\infty$ does not change the eigenvalues of the rotation matrix $\mathbf{R}$, namely $\{1, e^{\pm i\theta}\}$. Therefore, the rotation angle, θ, between views can be computed directly from the phase of the complex eigenvalues, $e^{\pm i\theta}$, of $\mathbf{H}_\infty$. The rotation angle θ, defined as the pure rotation feature $\mathbf{v}_r$, can be used to temporally align two videos with pure rotating cameras. In addition, the eigenvector of $\mathbf{H}_\infty$ corresponding to the real eigenvalue is the vanishing point, $\mathbf{v}_L$, of

(a) (b)

Figure 6.9: The computed rotation properties of two pairs of pure rotation shots, captured by the authors. (a) The rotation angle is $4.86°$, and $\mathbf{v}_L = [0.2011,\ 0.9796,\ 0.0001]^T$. (b) The rotation angle is $12.9°$, and $\mathbf{v}_L = [0.0273,\ -0.9996,\ 0.0001]^T$. The corresponding points are automatically found by the method described in [100].

the 3D rotation axis $\mathbf{L}$, since $\mathbf{L}$ is the unit eigenvector of $\mathbf{R}$ and

$$\mathbf{v}_L = \mathbf{KRK}^{-1}\mathbf{K}[\mathbf{I} \mid \mathbf{0}][\mathbf{L}^T\ 0]^T = \mathbf{KRL} = \mathbf{KL}. \tag{6.8}$$

The location of $\mathbf{v}_L$ can be used as the criteria to differentiate panning ($\mathbf{v}_L \approx [0\ \infty\ 0]^T$), tilting ($\mathbf{v}_L \approx [\infty\ 0\ 0]^T$), and swing ($\mathbf{v}_L$ is close to the image center). Two computation examples are shown in Fig. 6.9.

6.4.2 Pure Translation

In the pure translational motion case, the motion of the camera is a pure translation $\mathbf{t}_0$ with no rotation or change in the internal parameters. Geometrically, this is equivalent to the situation where the camera is static while points in 3-space move on straight lines parallel to $\mathbf{t}_0$. The imaged intersection of these parallel lines is the vanishing point in the direction of $\mathbf{t}_0$, which is also the epipole, $\mathbf{e}$, for both views since the camera centers and the point at infinity along $\mathbf{t}_0$ are collinear. It is known in [75] that, in this case, the fundamental matrix $\mathbf{F}$ between any two frames are equal and has a special form:

$$\mathbf{F} = [\mathbf{e}]_\times \mathbf{K}[\mathbf{I} \mid \mathbf{t}_0]\mathbf{P}^+ = [\mathbf{e}]_\times, \tag{6.9}$$

where $\mathbf{P}^+$ is the pseudo-inverse of $\mathbf{P}$, i.e. $\mathbf{P}\mathbf{P}^+ = \mathbf{I}$. It is evident that $\mathbf{F}$ has only two degrees of freedom, and can be determined uniquely from two point correspondences provided that the two 3D points are not coplanar with both camera centers. Consequently, the upper left 2×2 sub-matrix of $\mathbf{F}$ in Equation (6.6) degenerates to:

$$\mathbf{F}^{2\times 2} \sim \begin{bmatrix} 0 & -e_z \\ e_z & 0 \end{bmatrix}, \tag{6.10}$$

where e_z is the third element of the homogeneous epipole coordinate. Since Equation (6.10) provides no information on the camera ego-motion and also it is equal for any frame pair, it is impossible to make use of the fundamental ratios $\mathbf{v}_g$ in Equation (6.7) to align video sequences in the case of pure translation.

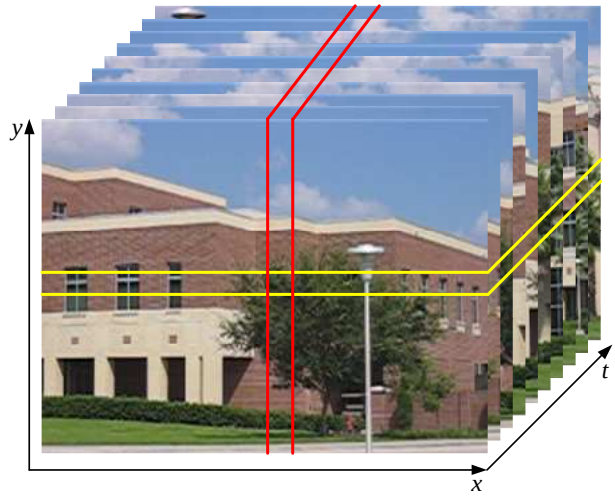
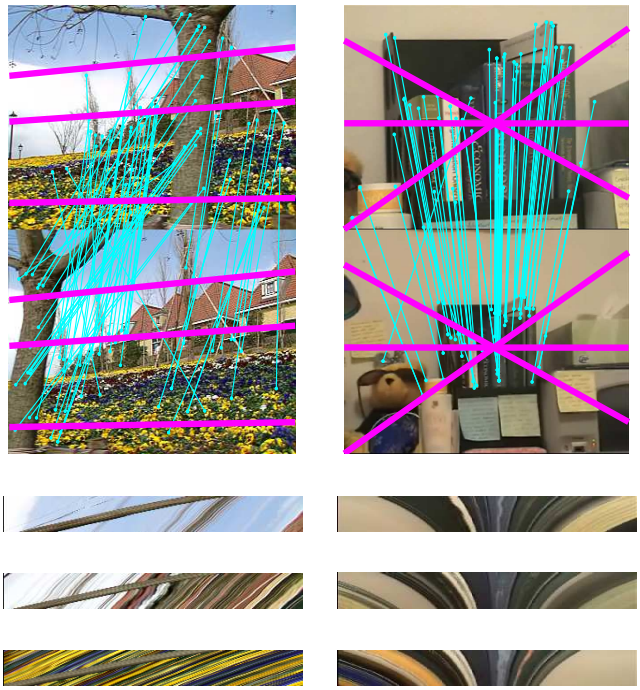


Figure 6.10: The 3D data volume (X: horizontal; Y: vertical; and T: temporal) and the two basic spatio-temporal 2D slices taken from the volume along the temporal dimension. A data volume is formed by putting together all the frames in a sequences, one behind the other. The two 2D slices are also known as X-T slices (yellow) and Y-T slices (red).

To quantize the camera ego-motion in the pure translational cases, we slice the three dimensional data volume (see Fig. 6.10). Traditionally, there are two ways to make the slices, i.e. X-T or Y-T as shown in Fig. 6.10, which reveal the temporal behavior usually hidden from the viewer, i.e. the X-Y slides or the frames. Each spatio-temporal slice is a collection of 1D scans in the same selected position of every frame as a function of time. Due to its effectiveness in exploring temporal events along a larger temporal scale, the spatio-temporal slice is widely used in human action detection and recognition [101, 96], mosaicing [188] and video representation [122]. However, in the previous efforts, the geometry behind the visual spatio-temporal slices are not fully explored.



(a) (b)

Figure 6.11: (a) Two frames of a pure translation shot. The computed epipole locates at $(0.9742, -0.2258, -0.0012)$. (b) Two frames of a zooming shot. (**Top**): the cyan lines connected the corresponding points computed by the method proposed by Lowe [100], while the magenta lines are the epipolar lines. (**Bottom**): The slices are cut from 3D volume along the three magenta epipolar lines to reveal the temporal behavior. Notice that the y-axis is the temporal direction.

We observe that the trajectories of the same 3D points are represented by two dimensional curves in the 2D slice images if we slice the data volume along the epipolar lines as shown in Fig. 6.11 (a). The first order derivative of the trajectory characterizes the relative translational speed. In the case of constant translational speed, the trajectories degenerate to straight lines, e.g. the tree branch in Fig. 6.11 (a). We define the relative translational magnitude, i.e. the relative

distance between the reference frame and the current frame along the x-axis of the slice shown in Fig. 6.11 (bottom), as the pure translational camera motion feature, $\mathbf{v}_t$. Note that our slices are different from the the traditional X-T slices [188] in that the projective distortion is considered here.

6.4.3 Zooming

In the case of zooming, the image effect can be approximated as a simple magnification [75],

$$\mathbf{x}' = \mathbf{K}'[\mathbf{I} \mid \mathbf{0}]\mathbf{X} = \mathbf{K}'\mathbf{K}^{-1}\mathbf{x}, \tag{6.11}$$

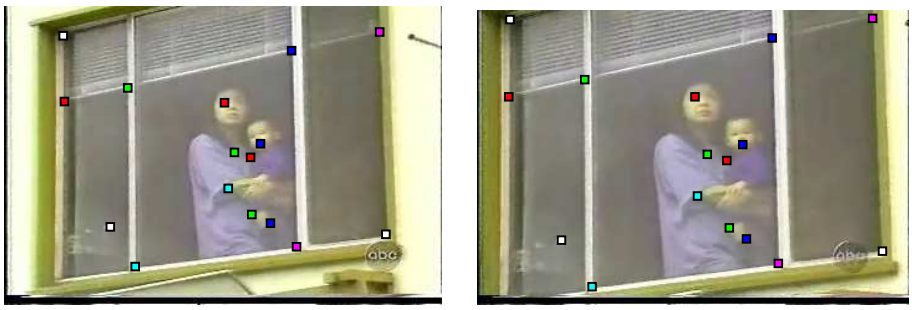
where we assume the zooming will not perturb the effective camera center. If the zooming only changes the focal lengths of $\mathbf{K}'$ and $\mathbf{K}$ then a short calculation show that

$$\mathbf{H}_A = \mathbf{K}'\mathbf{K}^{-1} = \begin{pmatrix} s\mathbf{I} & (1-\lambda)\mathbf{u}_0 \\ \mathbf{0}^T & 1 \end{pmatrix}. \tag{6.12}$$

where $\mathbf{u}_0$ is the inhomogeneous principal point, and $s = f'/f$ is the zooming factor. Therefore, the special form of the affine matrix $\mathbf{H}_A$ can be used to compute the zooming magnitude and the principal point. In the example of image pair in Figure 6.12,

$$\mathbf{H}_A = \begin{pmatrix} 1.1571 & -0.0069 & -28.3778 \\ 0.0167 & 1.1556 & -20.8401 \\ -0.0000 & -0.0000 & 1.0000 \end{pmatrix}. \tag{6.13}$$

In other words, $s = 1.16$, and the principal point locates at $(180.6, 132.7)$, close to the center of a 352×240 frame.



(a) (b)

Figure 6.12: Two frames of a zoom shot, from the ABC news. The corresponding points are automatically found by the method described in [100].

One interesting observation is that, since the effects of forward/outward translation and zooming are similar, the pure translational feature $\mathbf{v}_t$ can also be used to synchronize zooming sequences although it is not really a camera motion. For example, the two frames from a zoom out sequences, shown in Fig. 6.11 (b), can be treated as a outward tracking along a direction parallel to the principal axis. In the example of Fig. 6.11 (b), the translational moving speed (or the zooming factor) is not constant any more, and therefore the trajectories of the same 3D points are not as straight as those in Fig. 6.11 (a).

6.4.4 Camera Motion Characterization

From the above analysis, we conclude that two frames undergoing pure rotation, zooming and pure translation can be related by a general planar homography $\mathbf{H}$, an affinity $\mathbf{H}_A$ and a fundamental matrix $\mathbf{F}$ respectively. In the first two cases, the camera centers are fixed and $\mathbf{F}$ is undefined, and

Figure 6.13: Camera motion characterization results of a ABC news video feed. The first two rows denote the frame pairs of zoom shots. The middle two rows denote the frame pairs of panning shots. The bottom two rows denote the frame pairs of tilting shots.

thus the computed fundamental matrix is error-prone. In the last case, the extent of the motion depends on the inverse depth, and there is no single homography that satisfies all corresponding points.

We adopt the Bayesian model estimation method [169] to automatically select the best inter-frame relationship from the set $\{\mathbf{H},\ \mathbf{H}_A,\ \mathbf{F}\}$. After the shot motion is identified, the shot can be further classified. For example, we can discriminate a pure rotation shot as panning, tilting or swinging by checking the eigenvector of $\mathbf{H}$ corresponding to the real eigenvalue as described

in Section 6.4.1. The computed motion magnitudes, i.e. rotation angles, zoom magnitudes and relative translation, can be directly utilized for temporal alignments of shots with similar motions. Some characterized camera motions are illustrated in Fig. 6.13.

6.5 The Implementation Details of Alignment Algorithm

We assume that the given video sequences are known to have the similar camera ego-motion (the camera motion model can be characterized [204, 13] or automatically selected [169]).

6.5.1 Computing the Camera Ego-Motion Features

In this work, we concentrate on three camera ego-motion features: the fundamental ratios $\mathbf{v}_g$ in Equation (6.7) for general camera motion, the translational magnitude described in Section 6.4.2 for pure camera translation, and the rotation angles described in Section 6.4.1 for pure camera rotation. The initial frame-to-frame correspondences are accomplished using the SIFT feature proposed by Lowe [100]. The planar homography for pure rotation and fundamental matrix for general camera motion between the two views is computed using the MAPSAC (*maximum a posteriori sample consensus*) algorithm [169] that minimizes the reprojection errors, provided that there are sufficient point correspondences. The epipole of the pure translational shot is computed

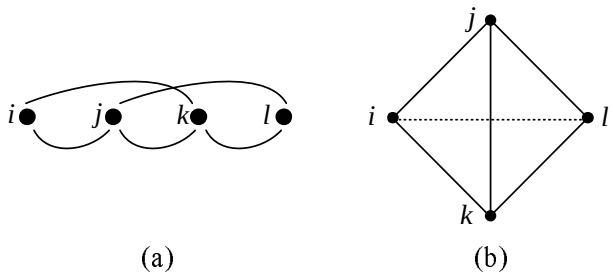


Figure 6.14: (a) Given four views and five of the C_2^4 interview fundamental matrices, we are able to compute the fundamental matrix between frame i and l which might have no overlapping areas. (b) The $4g5$ (four graph with five edges) is a solving graph [103].

using the eigenvalue decomposition of the matrix stacked by all lines connecting the corresponding points.

The above scheme assumes that there are nontrivial overlapping areas between the two frames of one video sequence. However, in the case of dynamic timeline model as discussed in Section 6.5.2, two frames i and l might be far away and hence have no correspondences for the computation of the fundamental matrices. In this case, the viewing graph theory [103] can be used for computing the fundamental matrices between the frames i and l, when there are two frames j and k which have overlapping areas with both i and l. In other words, the fundamental matrices inside two tri-views (i, j, k) and (j, k, l) are available as shown in Fig. 6.14.

In addition, the camera ego-motion, i.e. the relative translation and rotation, between the two views should be reasonably big to overcome noise. In this Section, we assume the corresponding relationship is simply a temporal shift $c(j) = j + \beta$. We can then readily compute the fundamental matrices between frames j and $j + \triangle t$, $j = 1 \ldots N - \triangle t$, for the video sequence $\{\mathcal{I}_j\}_1^N$. The

fundamental matrix $\mathbf{F}_{j,j+\triangle t}$ is meaningful provided that $\triangle t$ is significant. In our experiments, we choose $\triangle t$ as the number of frames for 1 sec. Note that the same $\triangle t$ should be used throughout all sequences. The solution to β can be simply recovered by an iterative point matching algorithm [198].

Finally, to increase the robustness of the proposed method, we use the coarse-to-fine framework since the synchronization in the coarser levels captures global features, and therefore an error in computation of frame correspondences will not be propagated to the rest of the warping path.

6.5.2 Timeline Model

In this Section, we consider the most general case where the timeline is dynamic. This problem can be formalized as follows: given a set of camera ego-motion features, i.e. $\{\mathbf{v}_j\}$ and $\{\mathbf{v}_{j'}\}$, estimate the dynamic matching indices, $j' = c(j)$ for the other set of input camera ego-motion features $\mathbf{v}_{j'}$, where $\mathbf{v}$ can be general motion $\mathbf{v}_g$, pure rotation $\mathbf{v}_r$, pure translational motion $\mathbf{v}_t$. The estimation is subject to the constraint that no frames displayed in the past can be captured in the future, which can be expressed as: for any given j_1 and j_2, if $j_1 < j_2$, then $c(j_1) \leq c(j_2)$. The error of alignment is computed incrementally using:

$$E(j, j') = dist(j, j') + \min\{E(j, j' - 1), E(j - 1, j' - 1), E(j - 1, j')\}, \tag{6.14}$$

where $dist(j, j')$ is computed using the mean squared error between $\mathbf{v}_j$ and $\mathbf{v}_{j'}$. Therefore, the synchronization determined by the set of parameters $c(j)$ is computed as

$$c(j) = \arg \min_{c(1) \leq \cdots \leq c(N)} \sum_{j=1}^{N} E(j, c(j)), \tag{6.15}$$

where N is the number of the input video frames. The optimization defined in Equation (6.15) is then solved using dynamic programming [20].

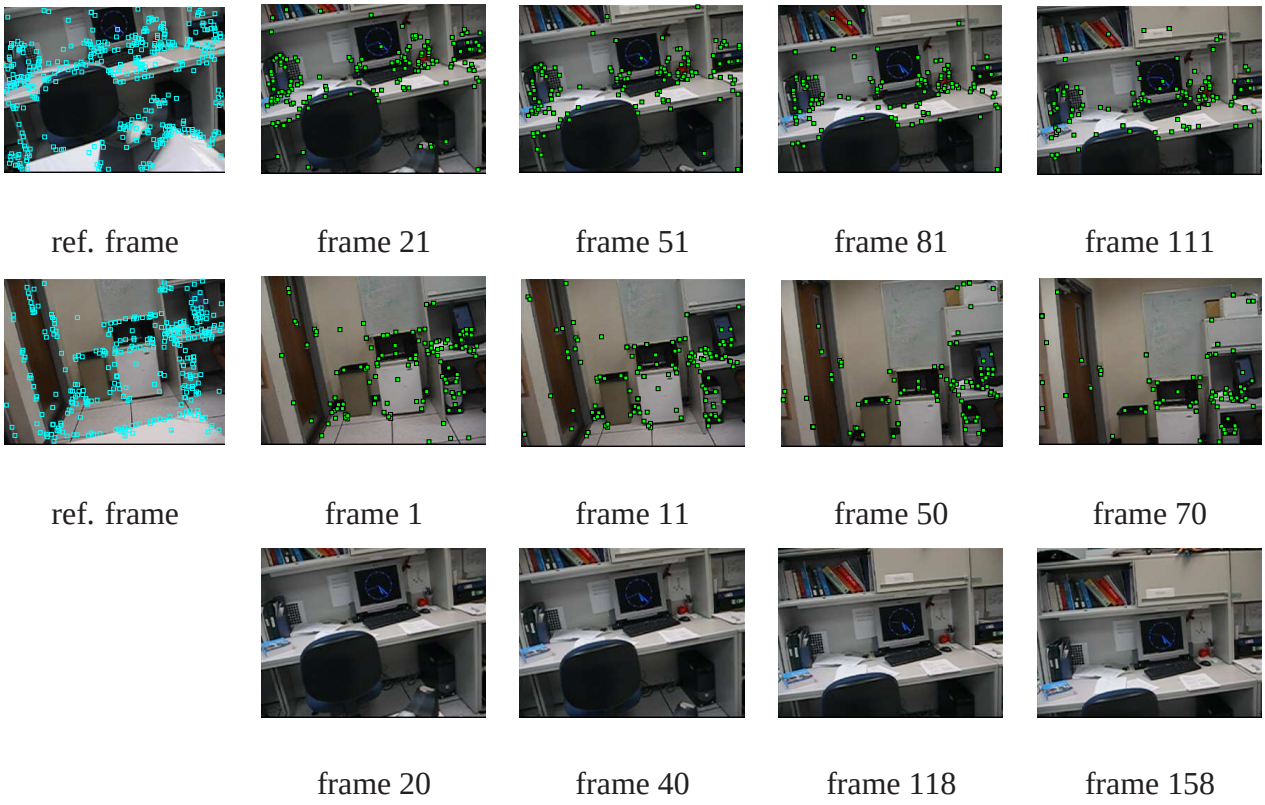


Figure 6.15: (**Top**) Sample frames from the first video, with extracted SIFT features superimposed, and the starting reference images (left). (**Middle**) Identical elements for the second video. (**Bottom**) Frames of the first video corresponding to the frames of the second video shown in the row above, according to our algorithm.

6.6 Experimental Results

In this Section, we demonstrate our method in cases where cameras are undergoing both general and special motions.

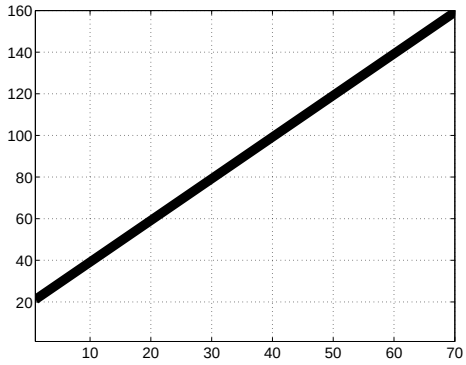


Figure 6.16: The estimated affine timeline model for the two video sequences in Fig. 6.15.

6.6.1 General Motion

As the first example, we took two sequences (70 frames and 160 frames respectively) of two non-overlapping indoor scenes. The trajectories of the two cameras were controlled by a CRS Plus robot to make sure they go through the similar motion. For this pair of sequences, we have ground truth information that the temporal dilation ($\alpha = 2.0$) of the two sequences by setting the ratios of the speeds of the robot arm motions as $2 : 1$ for the two sequences. Some example frames are shown in Fig. 6.15. The cameras are non-static with the main camera motion downwards. In this example, we manually match one pair of frames between two videos as shown in Fig. 6.15 left. The extracted SIFT feature points of the starting reference frame are superimposed on the reference images as cyan square markers, and the matched correspondences are shown as blue square markers in each frame. The estimated affine timeline model is $c(j) = 2.0014 * j + 18.1617$, as shown in the solid line in Fig. 6.16. The temporal dilation parameter, $\alpha = 2.0014$, accurately

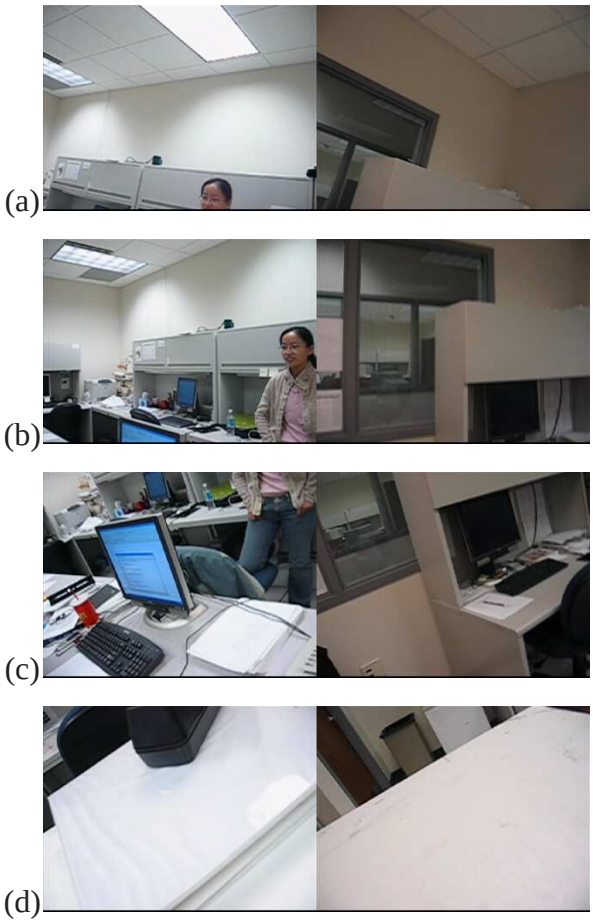


Figure 6.17: Four appended frames after synchronization, where the left (resp. right) half is from the first (resp. second) video.

matches the ground truth information. After applying the affine timeline, some corresponding frames in the first video sequences are demonstrated in Fig. 6.15 bottom.

In the second example, we take advantage of the available knowledge of the speed of camera motion, i.e. $2:1$. The input sequences have around 500 frames and 1000 frames (33 seconds), respectively. The two sequences are non-overlapping and have different lighting conditions as shown in Fig. 6.17. The two sequences are challenging in that some frames are very textureless

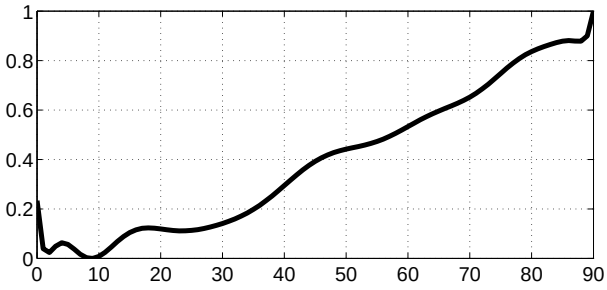


Figure 6.18: Average distance (rescaled between $[0, 1]$) of the camera general ego-motion features computed for different time delay.

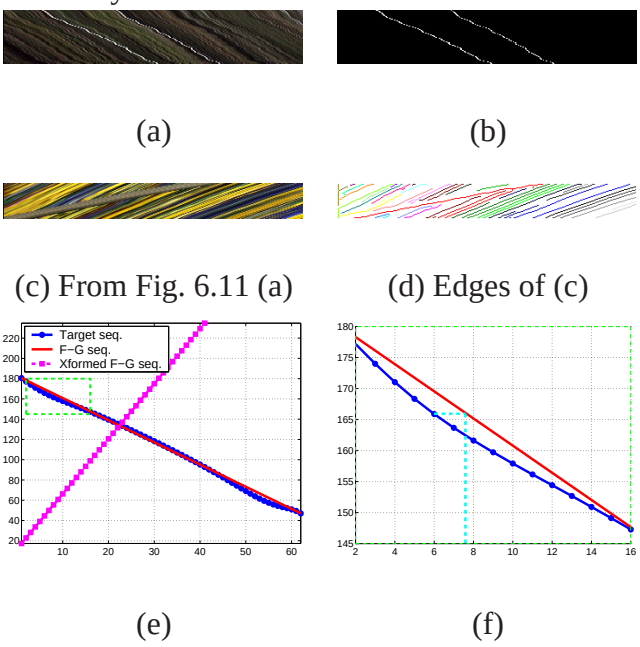


Figure 6.19: (a) The slice cut from the 3D volume along the cyan epipolar line in Fig. 6.20. (b) The two extracted curves of the foreground street lamps as marked by red arrows in Fig. 6.20. (e) The computed translational speed of the two sequences. (f) A close view of (e).

such as the frames temporally around (a) and (d), and that there are nontrivial moving chairs and talking person in the sequence in Fig. 6.17. Fig. 6.18 shows the average distance between the camera ego-motion features $\mathbf{v}_g$ as a function of the time delay. The graph goes to zero at 9, i.e. the

Figure 6.20: The top left is one frame from the source sequence (the flower garden sequence) while the top right is one frame from the target tree sequence. Four consecutive target frames superimposed with the foreground tree layers, which are extracted from the corresponding frames after synchronization using our algorithm.

time delay between the two video sequences. Corresponding frames based on this time delay and known time dilation 2 are shown in Fig. 6.17.

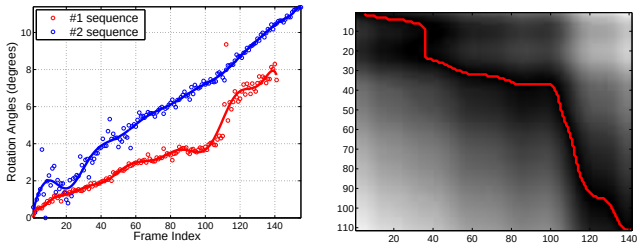
6.6.2 Pure Translation

To demonstrate our algorithm in the pure translational motion. We use the standard flower garden (F-G) sequence and the sequence (called the target sequence hereafter) from [194] as shown in

Fig. 6.20. Due to non-constant moving speed of the camera and the shaking, the slice cut from the 3D volume of the target sequence is not a straight line anymore as discussed in Section 6.4.2. To synchronize the two sequences, we used the left street lamp (the left curve in Fig. 6.19 (b)) and the tree branch (the curve with smallest slope in Fig. 6.19 (d)) to compute the relative translational magnitudes, which are illustrated in Fig. 6.19 (e). After inverse and scale the flower-garden curve, we have the red and blue curves. Evidently, we can't have a linear correspondence relationship between the sequences. For example, in the close view in Fig. 6.19 (f), the frame 6 of the target sequence should match frame 8 in the flower-garden sequence. After apply the dynamic programming, we have temporal alignment between the two sequences. To testify our results, we composited the layers of the foreground tree branch computed using the method proposed in [194] into the target background. Some of the frames are demonstrated in Fig. 6.20.

6.6.3 Pure Rotation

The last example is a pair of pure rotation sequences, captured by the authors, as shown in Fig. 6.22. The computed rotation angles using the method described in Section 6.4.1 are shown in Fig. 6.21 (a). To handle the computation errors, we fit a polynomial of degree 16 to the computed angles as solid curves. Fig. 6.21 (b) shows the similarity matrix between the two set of rotation angles. The value in this matrix is the absolute difference between pure rotation features, $\mathbf{v}_r^i$ and $\mathbf{v}_r^j$, where the superscripts i and j indicate the frame index of two video sequences respectively. The solid red



(a) (b)

Figure 6.21: (a) The computed rotation angles of two video sequences. (b) The similarity matrix and the time-warping path between two rotation sequences. The x-axis is the 1^{st} sequence (the lower one in (a)), while the y-axis is the 2^{nd} sequence.

line corresponds to minimum cost path through close-to-zero values. Based on the synchronization results, some corresponding frames from two sequences are appended in Fig. 6.22.

6.7 Conclusion

This chapter proposed a novel method to synchronize two video sequences of distinct scenes captured by cameras undergoing similar ego-motions. The proposed algorithm makes use of camera ego-motion features which are independent of the camera internal parameters, and therefore is able to synchronize videos captured by cameras with constant but different internal parameters. The strength of this method is that it can deal with arbitrarily moving camera and general scenes, as well as special camera motions. We demonstrated two common motions: pure translation and pure rotation. Other types of motions, e.g. zooming, can also be integrated into this framework

Figure 6.22: Four appended frames after synchronization, where the left (resp. right) half is from the first (resp. second) video.

provided that some feature vectors characterizing the camera ego-motions can be extracted from video sequences. The input of our algorithm is only the corresponding points.

In the cases when we know that the cameras have the same camera internal parameters, all elements of the fundamental matrices, rather than the four in Equation (6.7), should be utilized to improve the accuracy of the algorithm. In addition, in this work, the camera internal parameters

are assumed to be fixed throughout the video. We would like to relieve this constraint in the future work.

Our method has the following limitations. First, for a pair of pure rotation or translation shots, the capturing cameras must have fixed camera internal parameters, i.e. no zooming or focusing. On the other hand, in the case of pure zooming we assume that the zooming will not perturb the principal point. Second, in the computation of the simple camera motion magnitudes, the interval of frame pairs should be relatively big. Otherwise, the useful information is below noise level, e.g. eigenvector decomposition of $\mathbf{H}$ in equation (**??**) is not meaningful. Although, this limitation can be overcome in practice by taking every n frames, with n sufficiently large to ensure that the condition holds. Third, for the camera pose estimation, the current framework works well when the camera internal parameters are fixed. In the future work, we will extend the framework to deal with variable camera internal parameters using self-calibration method.

CHAPTER 7

3D VIDEO POST-PRODUCTION

In this thesis, we describe two different methods for video post-production: three-dimensional and two-dimensional. In cases where the source and target video sequences have similar camera motions, the foreground layer can be simply cutout from the source frames and pasted into the corresponding target frames, and therefore the two-dimensional approach is suitable. On the other hand, the two-dimensional method would introduce obvious temporal inconsistence and perspective distortions for non-similar camera motions in the source and target videos. As a result, the 3D information of the object needs to be extracted in this case. Although the 2D method is geometrically simple, the photometric effects such as shadows and reflections need to be considered for realistic results, which is discussed in the next Chapter 8.

This chapter in particular describes a video-based 3D approach to pull the alpha mattes of rigid or approximately rigid 3D objects from one or more source videos, and then transform them into another target video of a different scene, in a geometrically correct fashion. Our framework builds upon techniques in camera pose estimation (Chapter 4 and 5), 3D spatiotemporal video alignment (Section 6.2 in Chapter 6), depth recovery, key-frame editing, natural video matting,

and image-based rendering. Based on the explicit camera pose estimation, the camera trajectories of the source and target videos are aligned in 3D space. Combining with the estimated dense depth information, we significantly relieve the burdens of key-frame editing and efficiently improve the quality of video matting. During the transfer, our approach not only correctly restores the geometric deformation of the 3D object due to the different camera trajectories, but also effectively retains the soft shadow and environmental lighting properties of the object to ensure that it is harmonic with the target scene.

7.1 Introduction

In commercial film and television production, video matting and compositing operations make it possible for a director to transfer part of a scene between two video sequences. Currently, the film industry has more interest to transfer a part of 3D scene from one video to another, such that the audience could strongly feel the 3D effects after rendering and compositing. However, most of the past work on video matting and compositing focuses on the matting side and assumes that the camera poses in both the source and target scenes are the same or have a fixed 2D translation or scaling [3, 31]. Therefore, with simple temporal video alignment, an alpha blending is sufficient to composite the foreground object into the target view.

Therefore, these methods cannot handle situations where the cameras filming source and target videos have different motion. In such cases, these two videos must be aligned in a 3D space

(*a*) (*b*) (*c*) (*d*)

Figure 7.1: 3D object transfer between two videos for a "sit-talking" person. (a) Two original frames from the source video. (b) The naive transfer by a cut-paste (with an alpha channel) process without 3D alignment. It is clear to see that the augmentation is incorrect, where the pose and size of the person is not consistent with the target cameras and the sitting location is also changed. (c) Some distortion may happen during 3D transfer when an incorrect depth $d = 0$ is used (or called planar homography transfer). (d) Correct transfer when our estimated depth is used. *Note: our framework also allows some small non-rigid motion of the foreground object, such as hand moving.*

to reduce global geometric distortion and the 3D foreground object must be re-rendered from a different target viewpoint. If the foreground object is directly cut and pasted into the target view, the 3D appearance of the object may not be consistent with the target scene due to the different viewing trajectories and lighting conditions. Fig. 7.1.(b) shows a naive result with the cut-pasting

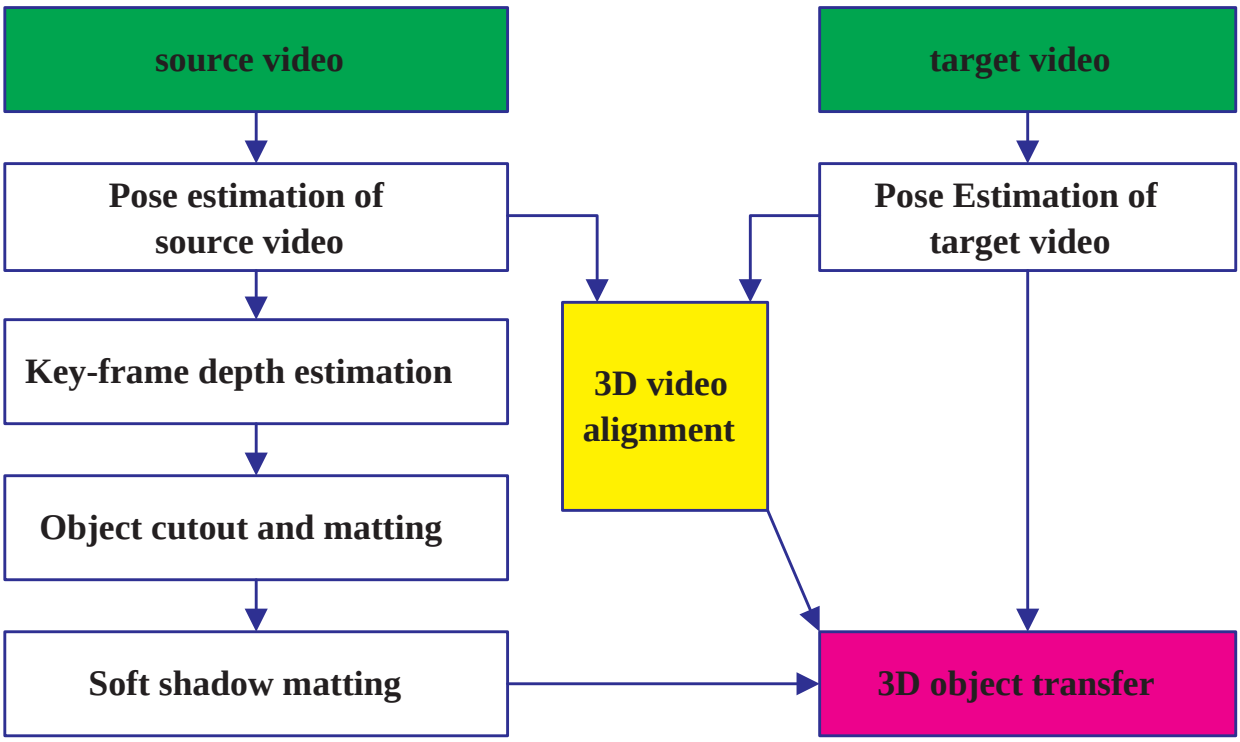


Figure 7.2: The flow chart of 3D video transfer process.

process by Chuang's video matting method [22], where the spatial artifact is clearly visible when the two videos have different poses.

To ensure the composited/transferred virtual objects are consistent with the existing objects in the target scene, our method explicitly estimates the poses of the cameras both for the source and target scenes, and aligns the videos along the computed camera trajectories in 3D space. Fig. 7.2 gives an overview of the 3D object transfer process. The inputs are two videos of different scenes acquired by two moving cameras: one is the source video with the foreground object, the other is the target video to receive the object. First, we estimate the camera poses for both the source and target video sequences. Next, the algorithm aligns these two videos along 3D camera trajectories

for the object transfer by minimizing the global differences between the source and transformed target viewing directions. In addition, after the pose estimation of the source video sequence, an initial depth of the foreground object is recovered, and further an intuitive graphical user interface (GUI) is designed to remove the depth errors due to small non-rigid motion or specular reflection. Then, we combine the depth information on video matting process to pull the high-quality alpha mattes of the object layer and shadow mattes of the shadow layer from the source video separately. Finally, using the optimal target viewing position estimated from the 3D video alignment, both layers are re-rendered and blended into the corresponding target frame. As a result, we not only correctly restore the perspective deformation of the object during the transfer, but also render the object which is spatiotemporally consistent with the target scene with realistic 3D effects as shown in Fig. 7.1.(d).

The proposed framework has the following contributions. First, we preform 3D spatiotemporal alignment given 2D imageries as input, and provide an effective solution to transfer 3D object between the videos of distinct scenes captured by two moving cameras, for which the existing methods would introduce noticeable unpleasing artifacts. Second, we utilize the 3D information of the source scene and design a depth driven GUI to effectively reduce the user interaction for object segmentation and alpha matting. Third, our approach is flexible to handle small non-rigid motion and specular reflection. Finally, the proposed approach is a fully lighting computation free algorithm, but is able to render a high-quality video with realistic environmental illumination. Our work not only advances the video matting and compositing problem from 2D to 3D direction, but

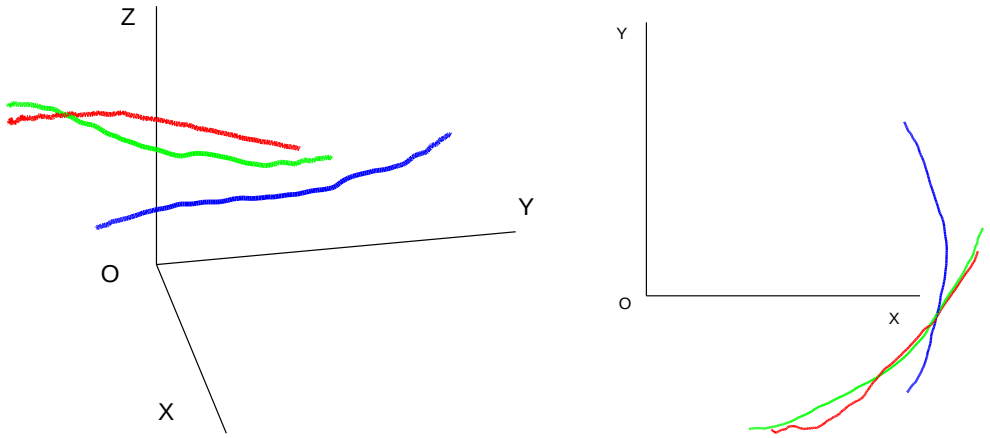


Figure 7.3: 3D camera trajectory alignment between the source “Beetle” and the target “flower-floor” sequences (see Fig. 7.13). The left is 3D view and the right side is top view. The red curve is the source camera trajectory. The blue and green curves are the target camera trajectories before and after alignment.

also provides a feasible alternative to the expensive camera control systems that have broadly been used in film industry.

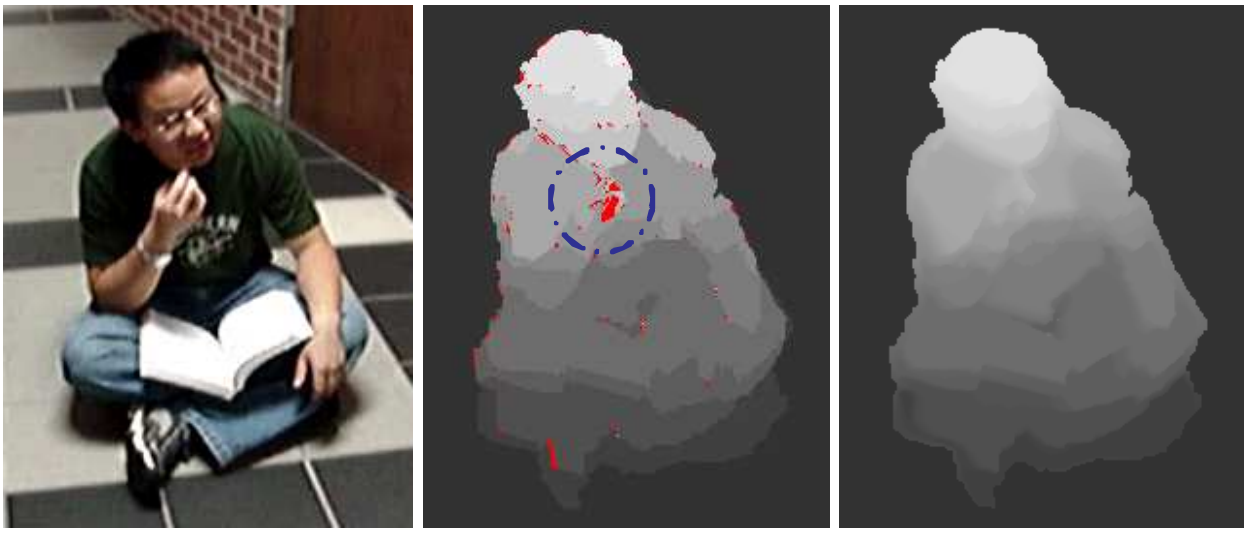
In this framework, the pose estimation is described in Chapter 4 and 5, and the 3D video alignment is presented in Section 6.2. Other steps are introduced in the remainder of this chapter as follows. Section 7.2 describes the depth estimation process and illustrate how to handle small non-rigid motion and specular reflection. Section 7.3 demonstrates alpha matting of the object and shadow layers, and the final rendering process. Then, several results are given in Section 7.4. Finally, we discuss the contributions and limitations of our approach, and provide the future direction in Section 7.5.

7.2 Key-frame depth estimation

Without the reference depth information, the object may not be correctly rendered, e.g. Fig. 7.1.(c). Therefore, a rough depth estimation or depth proxy is necessary for rendering the objects from a different viewing point [97, 9, 146, 193]. Currently, a number of stereo algorithms have been developed to recover the depth information from a stereo pair [158] or from a set of calibrated images [205, 57, 95]. Instead of using all of the video frames to compute depth of the scene, a set of key-frames are selected from the source video for the depth estimation to reduce the computational complexity. These selected key-frames cover the entire viewing field of the video with an approximately uniform gap. For example, in the"Sit-talking" sequence case (Fig. 7.1), we obtained 13 key-frames out of total 252 source frames with an approximate 3° viewing gap between the neighboring key-frames.

Using every five consecutive key-frames, an initial depth is computed by a multi-way cut algorithm which is adopted from a multi-frame segmentation framework [192]. However, the initial estimated depth (Fig. 7.4.(b)) has apparent discontinuities between the discrete depth labels since the multi-way algorithm is a labeling-based approach and the depth dimension is quantized into several labels (here we use $15 - 20$ depth labels). Moveover, from the results (Fig. 7.4.(b)), we can see that there are some apparent error around the non-rigid regions, such as the moving hand and the specular reflection regions[1]. Therefore, we need to refine the depth map to reduce the discontinuities and fix the depth errors for those non-rigid regions.

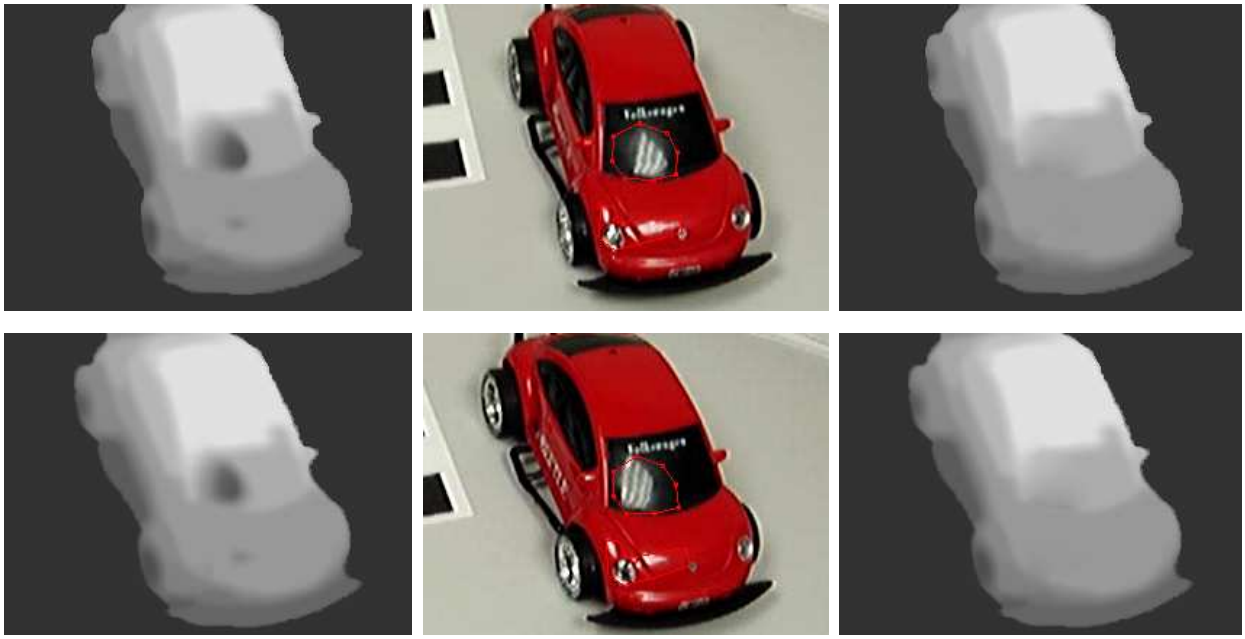
[1] Due to the fact that the specular reflection areas are view-variant, the corresponding motion of these regions is not consistent with the camera motion and its epipolar geometry. Here we call this observation is due to the non-stationary or non-rigid property of the specular reflection.



(a) (b) (c)

Figure 7.4: Depth refinement of non-rigid motion and occlusion. (a) One key-frame from the "sit-talking" sequence. (b) The initial estimated depth of the object using the multi-way cut algorithm. The red pixels have not been assigned a depth label due to the occlusion or non-rigid hand movement (inside the dotted green circle). (c) The refined depth map where the discontinuities are reduced and the unassigned pixels obtain a smooth depth by enforcing depth consistency constraint.

We first determine the highly discontinuous regions based on the gradients of the depth map and smooth those regions by a gaussian kernel. Then, we enforce a depth consistency constraint to ensure that the depth maps from different key-frames agree with each other. Intuitively, this means that given a pixel, p, with depth d in key-frame i, its 2D projection in key-frame j, $\pi_{j,i}(p, d)$, should have the same depth d. Therefore, the new depth of this pixel can be updated by iteratively



(*a*) (*b*) (*c*)

Figure 7.5: Depth correction of the view-variant specular reflections. (a) Depth before correction. (b) The feature points, where the feature at the bottom frame is directly projected from the top frame using the estimated depth. (c) Depth after correction.

enforcing the constraint as

$$D_i^{t+1}(p) = \sum_{i-1 \leq j \leq i+1} g(j-i) D_j^t(\pi_{j,i}(p, D_i^t(p))), \tag{7.1}$$

where $D_i(\cdot)$ is the depth map of frame i, $D_i^t(p)$ is the depth of pixel p in frame i at iteration t, $\pi_{j,i}(p,d)$ is the projection of pixel p with a depth d from key-frame i to key-frame j, and $g(\cdot)$ is a gaussian weight function. Fig. 7.4. (c) shows the refined depth map of one key-frame in the "sit-talking" sequence.

However, if the non-rigid region is large, the smoothness and consistency constraints may not fully eliminate the depth error. For example, the depth error due to the specular reflection is still visible at the windshield of the beetle as shown in Fig. 7.5. (a). To remove the depth error, one possible solution is to assume the depth of those regions is similar to the neighboring areas. Thus, after specifying the specular regions in the key-frame, a Poisson filling approach [133] is employed to enforce the boundary condition and smoothly propagate the depth information from the region boundary to the inside as shown in Fig. 7.5. (c).

For the next key-frame j, we project the feature points from the previous frame i to frame j using the projection function $\pi_{j,i}(p, D_i(p))$. If the projected regions correctly cover the specular reflection regions at frame j, the same Poisson filling process is applied to fix the error. Otherwise, the user can drag the feature points to relocate the reflection regions before fixing the error. Using the projected features, the user interaction can be dramatically reduced for the depth correction. Fig. 7.5 shows the feature projection and depth correction process. Once the correct depth map is obtained for each key-frame, we again use interpolation to generate depth map for the remaining non-key-frame by Eq. (7.3).

7.3 Video matting with soft shadow

Objects in natural scenes are often accompanied with some shadows, which have been ignored by the most previous video matting approach [22, 3]. In [30], Chuang et al. point out that shadow

element is essential for a realistic natural scene compositing. However, they only study the simplified case where scenes are illuminated by one dominant, point-like light source, and require a static camera. More importantly, they require that the relationship of the dominant light source, reference plane, and camera be matched in the source and target scenes. Here we will investigate a more general case where the shadow mattes vary in a wide range under multiple light sources.

7.3.1 Object cutout and matting

We decompose our matting problem into a two-layer system: shadow and object layers. However, it is not trivial to automatically separate these two layers, especially around the mixed or low contrast regions such as the bumpers of the Beetle. To separate these two layers, a user only needs to mark a few lines to specify the shadow region roughly along the boundaries in our GUI. Then, combining the background difference map with the specified feature points, a precise segmentation for the foreground object is obtained as shown in Fig. 7.6. (a). In the succeeding key-frame j (Fig. 7.6. (b)), these feature points are reused for the next frame segmentation by 3D projection, $\pi_{j,i}(p, d)$, as the depth correction. For the remaining frames, the foreground cutouts are interpolated from the neighboring key-frames using the 3D projection. Then, a trimap is created by small boundary expanding and shrinking, and the object matting is obtained as shown in Fig. 7.6. (c). If the object has a complicated silhouette (e.g., hair), a large unknown region may need to be specified in the key-frame using the GUI to estimate the object matting as shown in Fig. 7.7. Then, applying Poisson matting technique [150], the alpha mattes of the foreground object are computed.

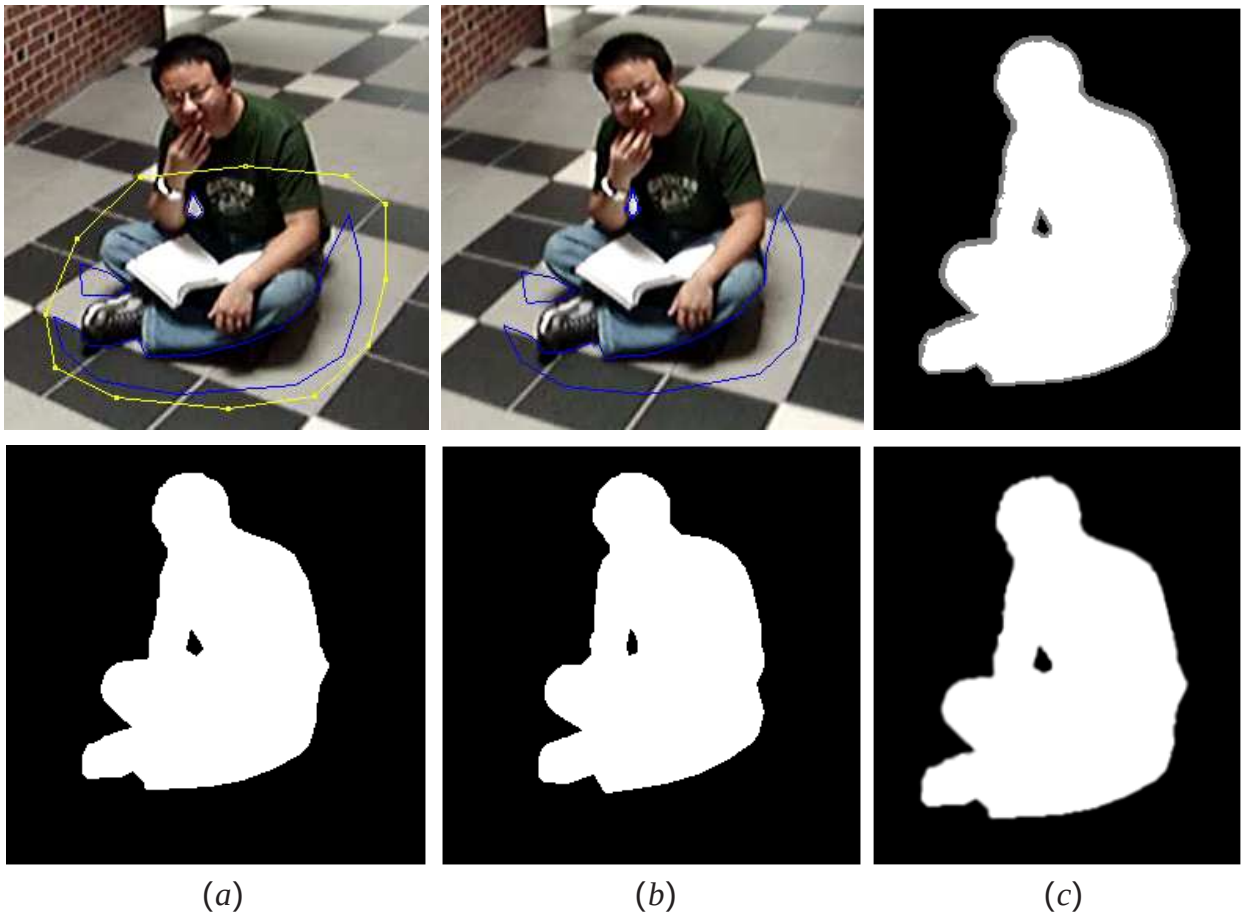


Figure 7.6: Object segmentation and alpha matting. (a) The first frame with two kinds of feature points. The blue curves are used to exclude shadow area from the foreground cutout, and the yellow curve is to mark a rough shadow region. (b) The top is the projected feature points at the succeeding key-frame using the estimated depth, the bottom is the corresponding foreground segmentation. (c) The trimap of the object layer (top) and its corresponding alpha mattes (bottom).

However, the consecutive alpha mattes may not be consistent along the temporal domain. Therefore, we again apply a gaussian kernel on each pixel p of alpha channel to ensure that the

Figure 7.7: Matting of complex objects. Left: A set of feature points in a key-frame of "doll" sequence, where two large unknown regions are marked by the green curves. Middle: The corresponding trimap. Right: The final matting result.

alpha mattes are coherent along the temporal domain.

$$\alpha_i(p) = \sum_{i-5 \le j \le i+5} g(j-i)\alpha_j(\pi_{j,i}(p, D_i(p))), \tag{7.2}$$

where $\alpha_i(\cdot)$ is the alpha map of frame i, and $g(\cdot)$ is a 1D gaussian weight function. After one pass temporal filtering, some boundary noise is removed particularly for the irregular boundaries, such as hair. Some refined matting results are shown in Fig. 7.8.

7.3.2 Shadow matting and editing

For the shadow layer, a rough shadow region needs to be marked in the first key-frame as shown by the yellow curve in Fig. 7.6. (a). Then a planar homography projects this shape to the other

Figure 7.8: Object matting refinement after the temporal consistency enforcement. Top: The results before the refinement. Bottom: The results after refinement. The left side is the results from "doll" sequence; the right side is results from "sit-talking" sequence.

frames to cutout the shadow boundary. Instead of using a trimap [24] to estimate alpha matting, we propose a new bi-map to extract the soft shadow matting. In this bi-map, there are only two parts: one is the definite background B and the other is the unknown regions U, which are separated by the specified region boundary. Given an estimated observed color I (Fig. 7.9. (b)) and lit color L without the shadow (Fig. 7.9. (c)), the initial shadow channel ρ^0 for each pixel can be estimated as

$$\rho^0 = \frac{I \cdot L}{\parallel L \parallel^2 + \varepsilon}, \tag{7.3}$$

where ε is a small constant to prevent zero division. However, this matting equation does not enforce the boundary condition where the alpha values at the shadow boundary, U, are 0. To enforce this boundary condition, we employ the $\nabla\rho^0$ as a guidance field with the membrane model

Figure 7.9: Shadow matting of one frame (Fig. 7.6.a) of "sit-talking" sequence. (a) The bi-map of the shadow layer. (b) After removing the foreground object, the observation I is estimated by Poisson filling. (c) The lit image L obtained from a reference image without this object. (d) The initial shadow matting ρ^0. (e) The enhanced shadow mattes using the membrane model.

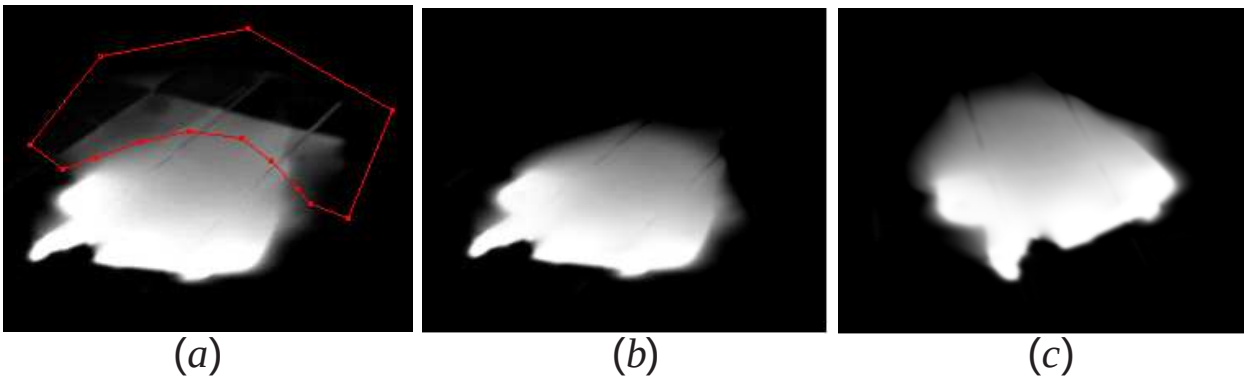


Figure 7.10: Local shadow matting editing. (a) The temporally smoothed shadow with the manipulating feature points. After marking the undesired shadow region (inside the red curves), we remove it and obtain the final result (b). (c) The shadow mattes warped into another frame.

to re-estimate the alpha value, ρ, such that

$$\min_{\rho} \int\int_{U} \parallel \nabla\rho - \gamma\nabla\rho^0 \parallel^2 \text{ with}(\rho|_{\partial U} = 0), \tag{7.4}$$

where $(\rho|_{\partial U} = 0)$ is to enforce the boundary condition, and γ is a constant coefficient which can be used to scale the guidance field. Using this approach, we not only enhance the alpha mattes by the guidance scale, but also effectively smooth the noise by the membrane model as shown in Fig. 7.9. (e). Once the shadow mattes are obtained for each frame, we further enforce the temporal consistency to refine the shadow mattes based on its planar homography invariant property, such that

$$\rho_1(p) = \frac{1}{N}\sum_{i=1}^{N}\rho_i(H_{1,i}p), \tag{7.5}$$

where $\rho_i(\cdot)$ is the alpha map of frame i, $H_{1,i}$ is the homography transformation from frame 1 to i with respect to the plane where shadow cast. Fig. 7.10. (a) shows the temporally smoothing shadow mattes. However, the smoothing results still have some errors due to the noise and imperfect estimated I_s. To achieve a desired shadow matting, we introduce a local editing process to allow users to locally manipulate the shadow matting by specifying a set of regions as shown in Fig. 7.10.

7.3.3 Layer compositing with image-based rendering

After preforming 3D camera trajectory alignment (Section 6.2), we have already determined the closest source view for each target frame. Then, using image-based rendering technique, we re-render the shadow and object layer with their alpha channels separately, and then composite them into the target view. The detailed steps are described as the follows.

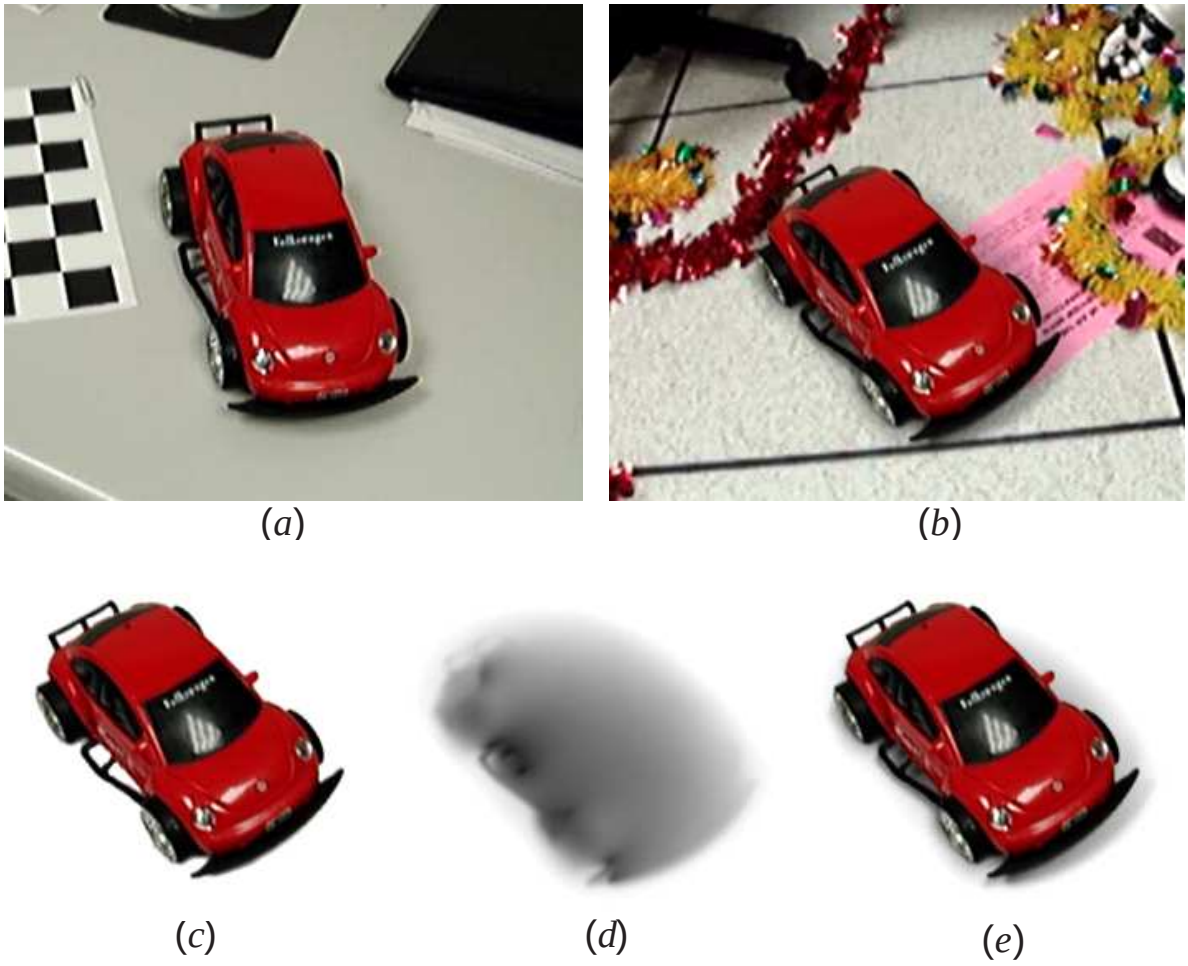


Figure 7.11: Layer compositing by image-based rendering. (a) The source frame. (b) The final result after compositing the rendered layers on the target view. (c) The rendering result of the object layer. (d) The rendering result of the shadow layer. (e) Layer compositing.

Given a new target view j, the nearest source frame i and its matting data are projected into the virtual view by projection $\pi_{j,i}(p, D_i(p))$ respectively. The results of both the shadow and foreground object are stored in separate buffers, each containing color, depth, and opacity. For each layer, a 3D mesh is created by converting their depth maps. Then, these two layers are

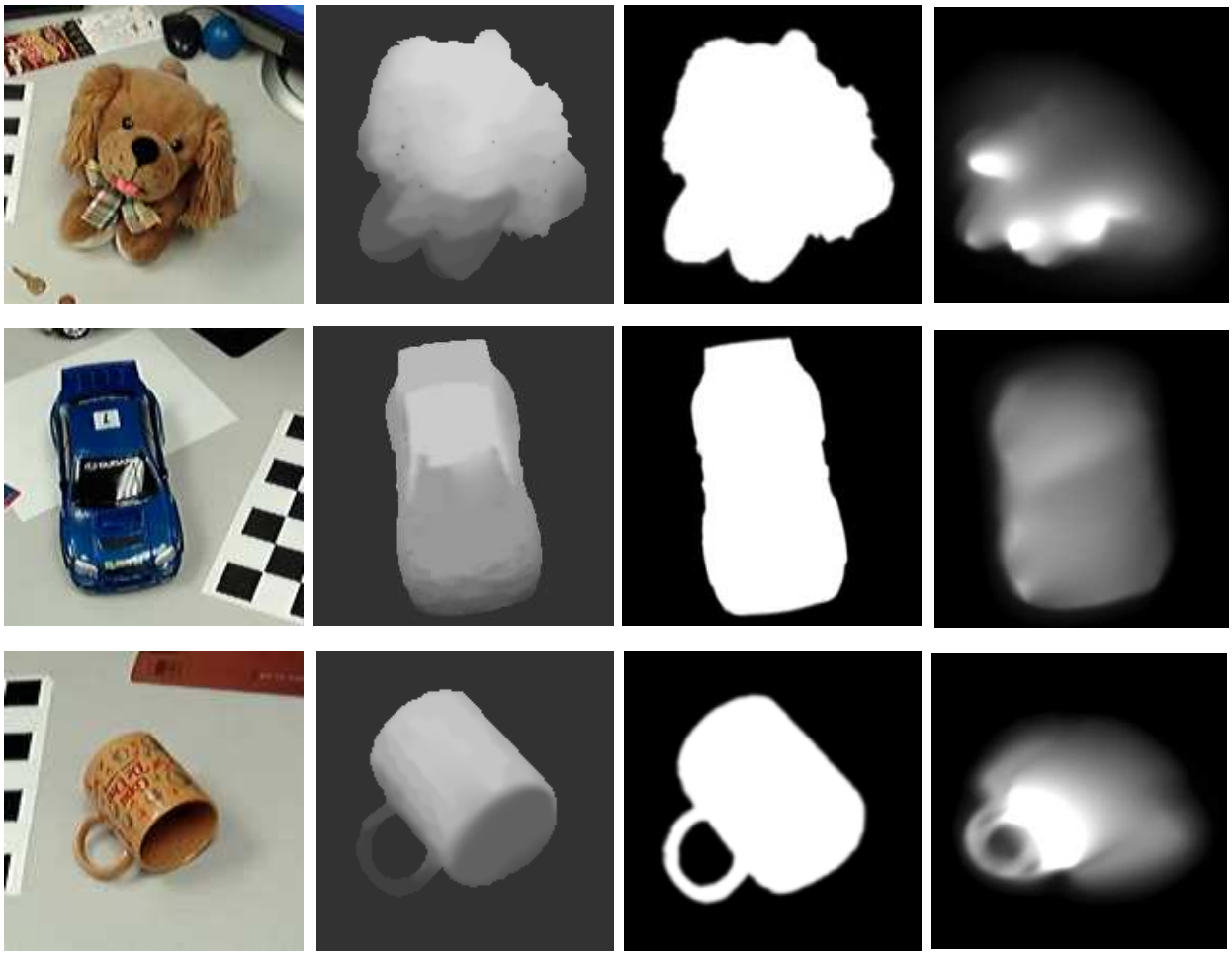


Figure 7.12: The sample depth estimation results in the other source video sequences. From left to right, the columns are source frames, depth maps, object mattes, and shadow mattes.

rendered and blended using the opacity channel to generate the final image as shown in Fig. 7.11. Even though there are some gaps (the viewing orientation difference $\beta_{i,j}$ between the closest P_o^j and P_s^i up to $10 - 15°$) between these two projection matrices P_o^j and P_s^i as shown in Fig. 6.3, our rendering shows very promising results due to the reliable depth estimation.

Table 7.1: The number of the key-frames used in our video sequences.

Name	Key-frame Number	Total Frame Number
Sit-talking	13	252
Beetle	20	405
Doll	21	433
Toy dog	17	400
Blue car	23	504
Mug	16	300

7.4 Experimental results

We have applied our approach to a number of video sequences, all of which are captured by a hand-held camera without using any assisting equipment. The resolution of all our video sequences is 640×480. To illustrate the soft shadow effects, our video sequences are taken from indoor scenes illuminated by multiple light sources.

Note that since the purpose of our approach is to transfer a 3D object from one video to another, we have more flexibility at the source side except that the environmental lighting condition is similar with the target scene. If the source video acquisition is controllable, we recommend using a relatively bright background to capture the the dim soft shadow of the object.

To test our approach, two groups of objects are selected. The first group contains the objects with highly specular materials, such as car and mug. The second group includes the objects with

soft and irregular hair or fur, such as doll, real human, and toy dog. Fig. 7.12 shows the sample frames of the source objects, the corresponding estimated depth, alpha mattes, and shadow mattes. The number of key-frames required for each sequence are given in Table 7.1.

With the strong support of the estimated depth map, our framework also provides the flexibility to change the target destination, T_t, and the rotation angle, θ, in a certain range, which allows the user to create some special effects, such as moving the objects or duplicating the objects in the target video.

Multiple Objects in One Scene: The first interesting application is to transfer multiple objects into one target video even though these objects are from different sources.

Object Moving, Colorizing, and Deforming: Due to the four degrees of freedom, we can translate, rotate, scale, or even deform the objects in the process of the video transfer.

Object Duplication: Another interesting application is duplicating the object into multiple copies, and each of them may have a slightly different appearance according to its own pose and destination.

Fig. 7.13 shows four synthetic video sequences to demonstrate the above applications. To feel the correct 3D visual effects. Using the proposed approach, we not only correctly recover the perspective deformation to lead a consistent and realistic 3D effect, but also implicitly explore lighting information which has been recorded in the source frames. Our approach avoids the challenges of the complicated lighting computation, and effectively restores the subtle variations of

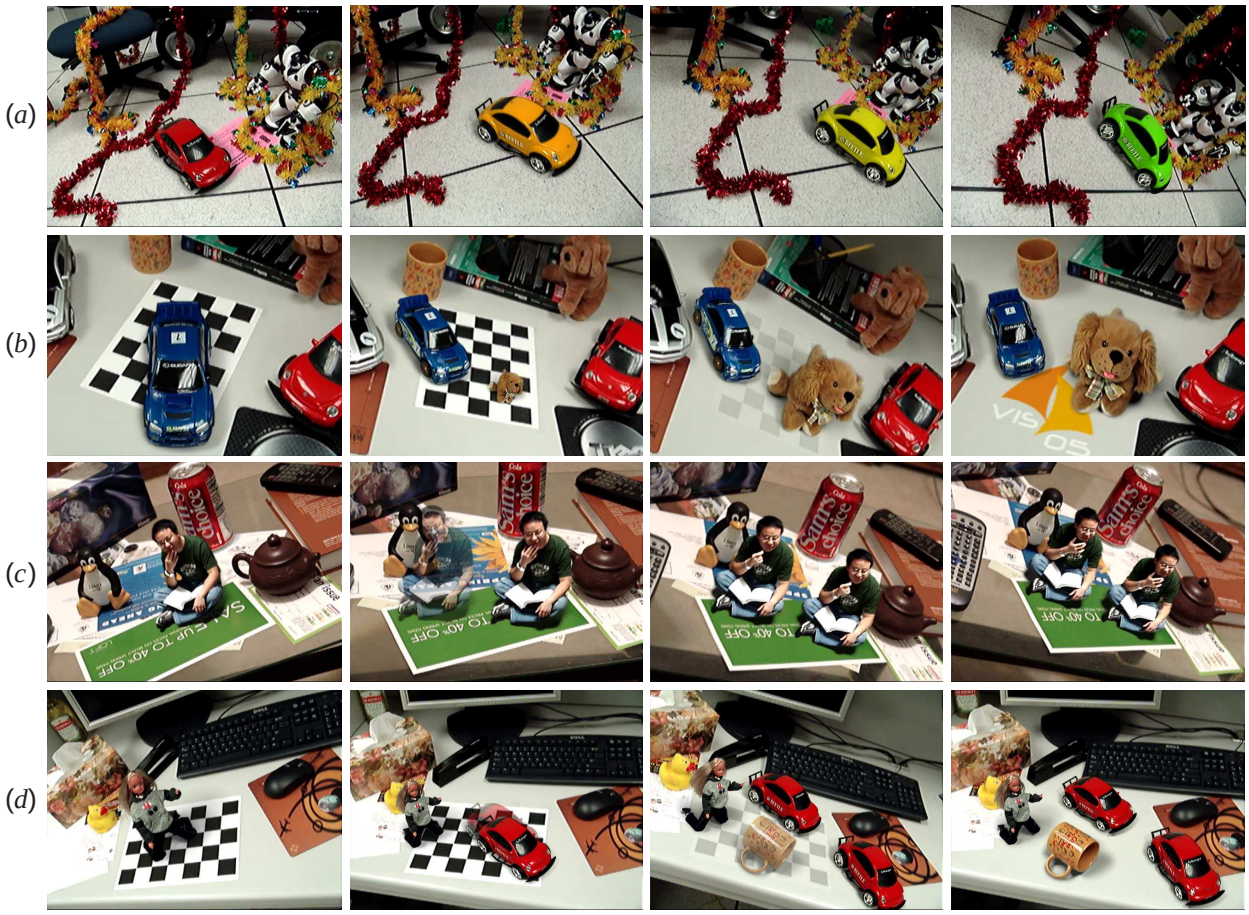


Figure 7.13: This figure shows four synthetic video sequences with one or more transferred objects from different source video sequences. (a) The Beetle is moving on the floor and the color is keeping change during the moving. (b) A blue car and a toy dog are transferred into the scene with an apparent scale change where the calibration grid is removed gradually. (c) The "sit-talking" person are duplicated during the transfer. (d) One Beetle is split into two with the quite different poses and locations, while the doll and mug are also correctly augmented into the target scene simultaneously.

the specular reflections to make the rendered objects to plausibly response the lighting environment of the target scene.

7.5 Conclusion and discussion

In this chapter, we have presented a new system for geometrically correct and visually realistic 3D object transfer that is capable of compositing 3D objects from the source videos into a distinct target video captured by a moving camera along a different trajectory. The proposed 3D spatiotemporal alignment approach provides a robust solution to align two non-overlapping videos of different scenes. With the assistance of our intuitive GUI, our approach is able to efficiently handle small non-rigid motion and specular reflection for the depth recovery, and also tackle the difficulties of mattes extraction for the object and its soft shadow. The experimental results strongly demonstrates that our approach is feasible to generate a realistic 3D video with a plausible environmental illumination from multiple video sources without expensive lighting computation.

While we have achieved a realistic and correct 3D video transfer between two different scenes, some limitations of our framework remain to be addressed. Transferring an object from one video to another is an extremely difficult problem for the general case. Here we put some constraints to simplify the problem and attempt to illustrate a solution. One constraint is that the transferred object is located at a plane where the shadow is relatively easier to be extracted and transferred. This constraint is consistent to most nature videos such as ground, sea plane, or man-made planar

surface. The second constraint is that we require the scene nearly static, which can provide more robust 3D information when the object is transferred between 3D scene. However, we also illustrate that our approach has some flexibility to handle small non-rigid motion of the objects, such as hand moving. For the large non-rigid motion such as human walking or running, the current framework is not working. One possible interesting solution is to extend our framework combining with the optical flow technique [22] for the non-rigid object transfer between 3D videos. Another limitation of our approach is that we need a sufficient amount of background texture or a wide field of view to perform camera calibration and pose estimation.

CHAPTER 8

2D VIDEO POST-PRODUCTION

The previous chapter describes a 3D method to solve the perspective distortions in the video post-production applications when the source and target videos are captured by moving cameras along distinct trajectories. Although the 2D method introduced in this chapter is relatively easier in geometric point of view, this chapter aims to relieve one of the earliest criticisms of linear perspective, found in Leonardo da Vinci's notebooks, that the liner perspective has inability to account for the atmospheric effects of light, haze and smoke. It is impossible to address all the natural phenomena in a single chapter. This chapter mainly concentrates on the shadow and reflection effects. It describes how geometrically correct and visually realistic shadows and reflections may be computed for objects pasted into target views.

8.1 Introduction

Compared to traditional compositing methods which either do not deal with the shadow or reflection effects or manually create the them for the composited objects, our approach efficiently

(a) (b) (c) (d) (e)

Figure 8.1: Sample result from our realistic compositing algorithm for shadows. Given an unknown video, e.g. a commercial video or a video taken by others, we recover the positions and orientations of the camera and light source. Then we place two video cameras at the camera and light source locations respectively. (a) is the view from the camera's viewpoint and (b) is captured by a camera at the light source's location and with its principal axis coinciding with the computed lighting direction. Using a traditional "faux shadow" method [187] by warping the foreground object's alpha matte (a) to create its shadow, we obtain a result in (c) . The results of our new shadow synthesizing method (d) compare favorably with the ground truth (e). Note the correct silhouette of the shadow map, visually convincing color characteristics of the shadow, and the seamless matte edges where the foreground and background shadows meet.

utilizes the geometric and photometric constraints extracted from a single target image to synthesize the shadows and reflections consistent with the overall target scene for the inserted objects. For shadows, we explore (i) the strong geometric constraints, i.e. the camera calibration and thus explicit determination of the locations of the camera and the light source; (ii) the relatively weaker geometric constraint, the planar homology, that models the imaged shadow relations when explicit camera calibration is not possible; and (iii) the photometric constraints that are required to match

the color characteristics of the synthesized shadows with those of the original scene. For each constraint, we demonstrate the working examples followed by our observations. In Figure 8.1, the results of our new shadow synthesis method (d) compare favorably with the ground truth (e), while (c) obtained by traditional "faux shadow" method is obviously fake.

Another challenge in video post-production applications is to generate realistic looking reflections of the inserted objects. Existing techniques either define beforehand a reflection model (e.g. [128]), or explicitly extract the reflection models from the images. The reflectance recovery algorithms, e.g. [12], typically either directly measure the reflectance on the object using a specific device or extract the reflectance from a set of images or a single image. While using a specific device is unlikely in many compositing applications, the reflectance recovery methods from images mostly limit to perfectly diffuse surfaces and require a 3D geometrical description of the surfaces of some object. Generally, synthesizing such reflections for compositing applications is inherently difficult because we are typically given only limited input, i.e. one view or a short video clip of a target scene. For example, suppose we want to insert a new synthetic object on the top of a real anisotropic mirror inside a real scene. This operation clearly requires taking into account the interaction between the new object and its environment (especially the mirror). This is impossible to do, if we do not have an approximation of the reflectance properties of the real surfaces in the image.

For reflection synthesis in this work, we focus on a slightly easier situation, where the target scene contains a ground plane and some up-right vertical objects, e.g. walls, crowds, desks, street lamps, trees, etc., which are common in both indoor and outdoor environments. Basically, we

divide the problem of synthesizing reflections of inserted objects into two subproblems. From the geometric point of view, we need to synthesize reflections which would be seen by the same camera used in capturing the target scene and be reflected by the true reflective media in the target scene. Physically, we aim to infer the most likely rendered reflections given the set of known reflection patches in the target view. This problem is formulated as a Maximum *A Posteriori* (MAP) estimation problem.

Our method has three advantages: first, it is a pragmatic approach since even in the single-view case, where the auto-calibration based on motion alone is impossible, it can exploit the ubiquitous presence of buildings, vertical objects and other man-made structures, which can often provide sufficient geometric constraints for the determination of the correct shadow location of an inserted object. Second, the framework is flexible in that various techniques suitable for different scenes can be easily integrated in it. Third, it is simple and easy to implement. To show the accuracy and the applications of the proposed method, we present the results for a variety of target scenes, including footage from commercial hollywood movies and 3D video games.

The rest of this chapter is organized as follows. We first discuss the synthesis of shadows in Section 8.2. Then, the method to synthesize the reflections are described in Section 8.3. Finally, Section 8.4 concludes this chapter with observations and proposed areas of future work.

8.2 Shadow Synthesis

This section describes the framework to synthesize shadows for objects cut from source videos and composited into the target videos. First, we explore the geometric constraints when the camera calibration is possible, and when it is not. Then,we describe the photometric constraint to match the color characteristics of the synthesized shadows to those of the original scene. Finally, we demonstrate the results of our method applied to various real images and applications to film post-production.

8.2.1 Geometric Constraints

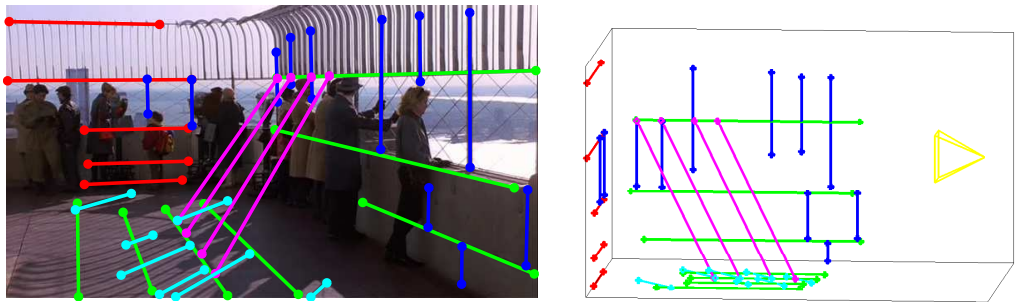
We first describe how to determine the position and orientation of the light source and camera when the imaged scene structure provide enough calibration constraints, which is called “strong geometric constraints”. Then we show that it is still likely to generate the shadow of an inserted object, provided that it is planar or distant, when the camera calibration is not possible. We refer to this constraint as “weak geometric constraint”. For each case, we give some working examples followed by discussions.

8.2.1.1 *Strong Geometric Constraints*

The calibration methods presented in Chapter 4 can be used for camera calibration and light source orientation estimation. For example, given a typical frame (Figure 8.2 (a)), we are able to extract three sets of parallel lines for camera parameter estimation shown in red, green and blue respectively. Therefore, by intersecting the image projections of the lines along each direction, we can compute the three mutually orthogonal vanishing points, denoted by $\mathbf{v}_x$, $\mathbf{v}_y$ and $\mathbf{v}_z$, along the world X-axis, Y-axis and Z-axis, respectively. The three mutually orthogonal vanishing points together with other constraints that can be obtained from the priors of a normal camera (e.g. zero skew, $\gamma = 0$, and known aspect ratio λ, usually 1 (square pixels), 1.2 (widescreen) or 0.9), are sufficient to solve for the five unknowns of the image of the absolute conic $\boldsymbol{\omega}$ (see Section 3.1). After the camera calibration, we can compute the camera internal and external parameters, i.e. the projection matrix $\mathbf{P}$ in Equation (3.1), up to a scale as shown in [40]. The scale is related to the camera location, and can be eliminated as follows. The fourth column, $\mathbf{p}_4$, of $\mathbf{P}$ is nothing but the projection of the 3D world origin, $[0\ 0\ 0\ 1]^T$, since (denoting the i^{th} column of $\mathbf{P}$ as $\mathbf{p}_i$), one can show that

$$[\mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3\ \mathbf{p}_4][0\ 0\ 0\ 1]^T = \mathbf{p}_4.$$

Therefore, given the image of the world origin, and by taking the length of a vertical object as unit distance, one can remove the scale ambiguity. For instance, in Figure 8.2 (a), without loss of generality, we can use the corner of the visible walls, at image location $(313.4, 206.4)$, as the imaged world origin in which case $\mathbf{p}_4 = \alpha[313.4\ 206.4\ 1]^T$, where α is the similarity ambiguity. If we now specify the height of the upper most green line from the ground plane as the unit distance,



(a) (b)

Figure 8.2: (a) One frame of the movie "Sleepless in Seattle" (1993) with the extracted feature lines plotted by five different colors: red, green and blue lines are along X, Y, and Z directions respectively in 3D world coordinate system, cyan lines are shadow lines of some spiked fence surrounding the 86^{th}-floor observation deck of the Empire State Building, and the magenta lines are along the light source direction. (b) The recovered $3D$ scene from a different viewpoint other than the original camera. The $3D$ feature lines are plotted in the same colors as those in (a), and the yellow square pyramid on the right indicates the original camera location.

we can remove the unknown similarity ambiguity α, and hence the compute the camera location as $(1.83, 8.13, 0.96)$.

A partially reconstructed scene of the image in Figure 8.2 (a) is shown in Figure 8.2 (b). Note that we do not need to fully reconstruct the scene of the target scene for image compositing applications since it is already present in the existing image. Note also that we reconstruct the scene in Figure 8.2 (b) using the planar homographies that map the world planes to the image planes, since the traditional triangulation method [74] would not work in our case due to the fact that there

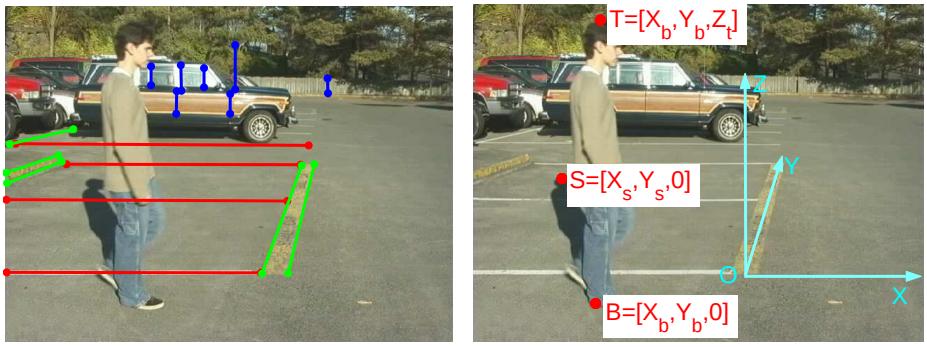
might be only a single view available. For example, we compute the planar homography, $\mathbf{H}_z$ that maps the world plane $Z = 0$, i.e. the ground plane, to the image plane as $\mathbf{H}_z = [\mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_4]$.

The imaged light source position, $\mathbf{v}$, can be computed by intersecting the images of the parallel lines along the light source direction, e.g. the magenta lines in Figure 8.2 (a). Note that $\mathbf{v}$ is not visible in Figure 8.2 (a). However, since the target scene in Figure 8.2 is illuminated by an infinite light source (the sunlight), the orientation of the light source the angles ϕ_x (respectively ϕ_y and ϕ_z) between the light source direction and the world X axis (respectively Y and Z axis) can be computed by (see also Section 4.2.4),

$$\phi_j = \cos^{-1} \frac{\mathbf{v}_j^T \boldsymbol{\omega} \mathbf{v}}{\sqrt{\mathbf{v}^T \boldsymbol{\omega} \mathbf{v}} \sqrt{\mathbf{v}_j^T \boldsymbol{\omega} \mathbf{v}_j}}, j \in \{x, y, z\}. \tag{8.1}$$

For the image in Figure 8.2, the computed angles are $\phi_x = 119.6°$, $\phi_y = 138.9°$ and $\phi_z = 64.4°$.

The second computation example is the single view shown in Figure 8.3, available from University of Washington. The same process for the camera calibration and the computation of $\mathbf{P}$ described above can be simply applied using the feature lines in Figure 8.3 (a). The difference is that Equation (8.1) can not be used to compute the light source orientation, since it is difficult to identify more than one line along the light source direction. Consequently, it is unlikely that we can compute the imaged light source, $\mathbf{v}$, from a single view shown in Figure 8.3. In this case, however, we still can compute the light source orientation by using two feature points along the lighting direction: $\mathbf{t}$ on the person's head and its cast shadow position $\mathbf{s}$ on the ground shown in Figure 8.3 (b). Without any difficulty, we can identify the corresponding bottom point $\mathbf{b}$ of $\mathbf{t}$ on the ground plane (the XY-plane), such that it has the same 3D X and Y coordinates as $\mathbf{t}$. Then, we



(a) (b)

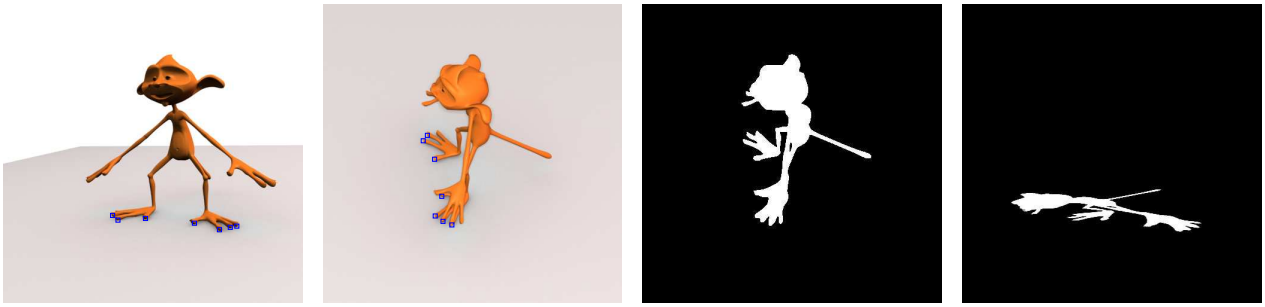
Figure 8.3: (a) A view with the extracted feature lines for camera calibration plotted in three different colors, each of which has the same definition as that in Figure 8.2. (b)The 3D world coordinate system and the 3D coordinate values, denoted by corresponding upper-case characters, of three image feature points ($\mathbf{t}$, $\mathbf{b}$ and $\mathbf{s}$) used to compute the orientation of the light source.

compute the points, $\mathbf{b}$ and $\mathbf{s}$, in the world coordinate system as

$$[X_b\ Y_b\ 1]^T \sim \mathbf{H}_z^{-1}\mathbf{b}, \quad [X_s\ Y_s\ 1]^T \sim \mathbf{H}_z^{-1}\mathbf{s}.$$

The Z coordinate, Z_t, of the feature point $\mathbf{t}$ is then estimated from the equation (two equations, one unknown), $\mathbf{t} \sim \mathbf{P}[X_b\ Y_b\ Z_t\ 1]^T$. For the distant light source, the sun, two 3D points, $\mathbf{t}$ and $\mathbf{b}$, along the lighting direction are enough to give us light source direction as $[X_b - X_s,\ Y_b - Y_s,\ Z_t]^T$.

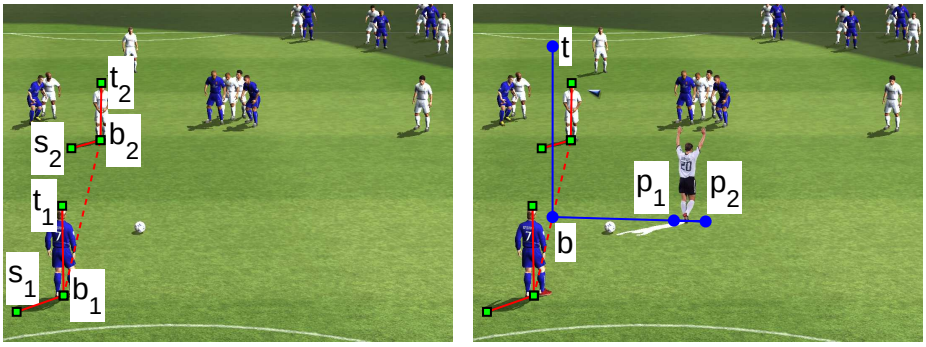
Consequently, as shown in Figure 8.4, the warping from the alpha matte (Figure 8.4 (c)) of the object (Figure 8.4 (b)) captured from the light source to the background (Figure 8.4 (a)) can be described as a planar homography $\mathbf{H}$, that can be computed using the corresponding feature points on the background plane of the two images (Figure 8.4 (a) and (b)). We then warp (Figure 8.4 (c)) to create foreground object's shadow matte (Figure 8.4 (d)).



(a) (b) (c) (d)

Figure 8.4: (a) and (b) are two views defined in Figure 8.1 (a) and (b). (c) is the alpha matte of (b). To warp the alpha matte (c) viewed from the light source direction into the final shadow map (d), we need to compute the displacement map. When the background is planar, the displacement map can be described as a planar Homography $\mathbf{H}$, which can be computed using the corresponding feature points on the background plane of the two images (a) and (b).

Before we end this section, we would like to make a few observations. First, it might be difficult to approximate the camera pose and lighting direction by simply looking at shadow lines in a perspective view. For example, it is difficult to tell the angleβ in 3D world between the cyan shadow lines (Figure 8.2 (a)) and the red lines, since the imaged shadow lines intersect at a finite point, $(678.8, 82.5)$, and hence each of them gives different values of that angle β. However, our method is able to compute β as $56.3°$. Second, single view calibration is necessary in most cases, even for clips from commercial movies, since static shot is the most basic camera shot in all motion pictures. The last remark is that the new findings of camera calibration objects, such as parallel shadows [40] and co-planar circles [45], could be easily integrated into this framework although we do not give examples of the latter here.



(a) (b)

Figure 8.5: Example to find the correct shadow pixels of an inserted real soccer player into a snapshot of the game FIFA 2003. (a) shows three pairs of corresponding points under a planar homology, i.e. $\langle \mathbf{t}_2, \mathbf{s}_2 \rangle$, $\langle \mathbf{t}_1, \mathbf{s}_1 \rangle$ and $\langle \mathbf{b}_1, \mathbf{b}_1 \rangle$. The point $\mathbf{b}_2$ is used for the computation of vertical vanishing point, but not used for computing $\mathbf{H}$. (b) plots the computed shadow pixels of the inserted object marked in white.

8.2.1.2 *Weak Geometric Constraints*

For the cases where camera calibration and the explicit geometric light source estimation are not possible, we utilize a relatively weak constraint, the planar homology, which makes it possible to synthesize the correct shadows of an inserted object if the object is planar or distant.

One example target scene is shown in Figure 8.5 (a), and the computation process is as follows. We first choose three pairs of corresponding points under a planar homology, e.g. $\langle \mathbf{t}_2, \mathbf{s}_2 \rangle$, $\langle \mathbf{t}_1, \mathbf{s}_1 \rangle$ and $\langle \mathbf{b}_1, \mathbf{b}_1 \rangle$ as input. Then, we compute the planar homology, $\mathbf{H}$, directly as [75]:

$$\mathbf{H} = \mathbf{I} + (\mu - 1)\frac{\mathbf{v}\mathbf{l}^T}{\mathbf{v}^T\mathbf{l}}, \tag{8.2}$$

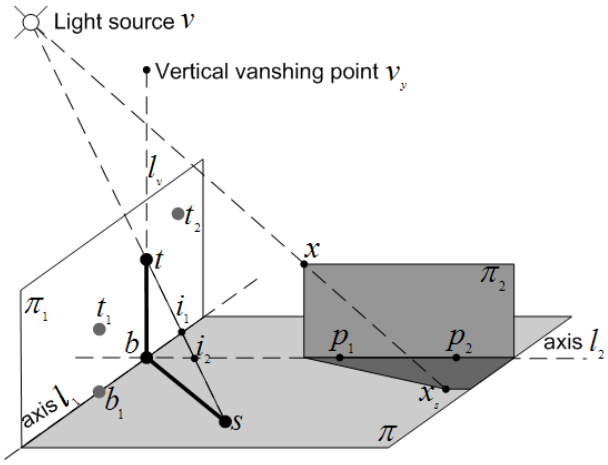


Figure 8.6: The geometry for computing the shadow positions for any points on plane $\boldsymbol{\pi}_2$, given the planar homology, $\mathbf{H}$, that relates the planar object, $\boldsymbol{\pi}_1$, and its shadow (cast on a ground plane $\boldsymbol{\pi}$).

where $\mathbf{I}$ is the identity matrix, μ, $\mathbf{v}$ and $\mathbf{l}$ are given by

$$\mathbf{v} = (\mathbf{t}_2 \times \mathbf{s}_2) \times (\mathbf{t}_1 \times \mathbf{s}_1),$$

$$\mathbf{l} = ((\mathbf{t}_2 \times \mathbf{t}_1) \times (\mathbf{s}_2 \times \mathbf{s}_1)) \times \mathbf{b}_1,$$

$$\mu = \{\mathbf{v}, \mathbf{t}_1; \mathbf{s}_1, \mathbf{i}_1\}.$$

Note that $\mathbf{i}_1$ is the intersection in the image plane, although the light ray $\mathbf{t}_1\mathbf{s}_1$ and the axis, $\mathbf{l}$, are unlikely to intersect in 3D world. Now, we have a planar homology, $\mathbf{H}$, which is the transformation between the image of a planar object, i.e. the plane $\boldsymbol{\pi}_1$ determined by the three points ($\mathbf{t}_2$, $\mathbf{t}_1$ and $\mathbf{b}$), and the image of its shadow on the plane $\boldsymbol{\pi}$.

Next, we describe how to determine the correct shadow position of the inserted real player in Figure 8.5 (b), which is matted from Figure 8.12. The distant standing person can be approximated

as a planar object in some vertical plane, denoted by $\boldsymbol{\pi}_2$ as shown in Figure 8.6. Since we have already computed the planar homology, $\mathbf{H}$, from the plane $\boldsymbol{\pi}_1$ to $\boldsymbol{\pi}$, we know its vertex $\mathbf{v}$ and axis $\mathbf{l}_1$. The new homology, $\mathbf{H}_2$, that maps the points on the plane $\boldsymbol{\pi}_2$ to their shadows on plane $\boldsymbol{\pi}$, has the same vertex $\mathbf{v}$ as that of $\mathbf{H}$, and its axis, $\mathbf{l}_2$ is the intersection of the plane $\boldsymbol{\pi}_2$ and $\boldsymbol{\pi}$, which can be found by specifying two points $\mathbf{p}_1$ and $\mathbf{p}_2$ on the intersection, i.e. $\mathbf{l}_2 = \mathbf{p}_1 \times \mathbf{p}_2$. The manually specified points are shown in Figure 8.5 (b). Denote by $\mathbf{b}$ the intersection of the two axes $\mathbf{l}_1$ and $\mathbf{l}_2$, and by $\mathbf{v}_y$ the vertical vanishing point. $\mathbf{v}_y$ can be computed by intersecting two vertical objects, such as $\mathbf{t}_2\mathbf{b}_2$ and $\mathbf{t}_1\mathbf{b}_1$ in Figure 8.5 (a). We then randomly choose one point $\mathbf{t}$ on the vertical line $\mathbf{l}_v$ passing through $\mathbf{b}$ and $\mathbf{v}_y$. Finally, $\mathbf{H}_2$ is computed as:

$$\mathbf{H}_2 = \mathbf{I} + (\mu_2 - 1)\frac{\mathbf{v}\mathbf{l}_2^T}{\mathbf{v}^T\mathbf{l}_2},$$

where

$$\mu_2 = \{\mathbf{v}, \mathbf{t}; \mathbf{H}\mathbf{t}, (\mathbf{t} \times (\mathbf{H}\mathbf{t})) \times \mathbf{l}_2\}.$$

Note that the four points should be scaled to inhomogeneous coordinates to compute μ_2. Consequently, given any point, $\mathbf{x}$, in the plane $\boldsymbol{\pi}_2$, it is easy to compute its shadow position, $\mathbf{x}_s$, on the plane, $\boldsymbol{\pi}$, by simply applying the 2D transformation, $\mathbf{x}_s \sim \mathbf{H}_2\mathbf{x}$. The computed shadow pixels for the inserted object in Figure 8.5 (b) are marked as white.

The technique described in this section is mostly related to the popular technique, commonly called “faux shadow” in the film industry [187], for which artists also use the foreground object’s alpha matte to create its shadow by warping or displacement-mapping the shadow. However, compared to “faux shadow” created by hand, the proposed approach has two advantages. First,

our method models the imaged shadow relations by the planar homology and is able to obtain geometrically correct shadow positions, while the geometric accuracy of the "faux shadow" highly depends on the experience of the compositor. Second, our method infers the most likely rendered shadow colors from the existing shadows in the target scene, while color characteristics of the "faux shadow" are manually adjusted by the compositor. This photometric constraint is described in the next section. In addition, the proposed method is simple and easy to implement. The major interactions consist of specifying three pairs of points $\langle \mathbf{t}_2, \mathbf{s}_2 \rangle$, $\langle \mathbf{t}_1, \mathbf{s}_1 \rangle$ and $\langle \mathbf{b}_1, \mathbf{b}_1 \rangle$ as shown in Figure 8.5 (a), and two points $\mathbf{p}_1$ and $\mathbf{p}_2$ as shown in Figure 8.5 (b). Notice that the above given examples are for vertical objects, since we observe that vertical objects are ubiquitous in the real world, and also typically, we are interested in inserting a new actor into the target scene, which is often orthogonal to a reference plane.

8.2.2 Photometric Constraints

While the geometric constraints described in the previous section help us to place shadows at correct positions, we also need to match the color characteristics of the shadows to those of the target scene. To create visually realistic shadows in the target image, we enforce the shading image values [21] of the synthesized shadows of the inserted objects to be the same as those of the shadows cast by the existing objects in the target scene. The shading image (or illumination image), $\mathbf{S}(x, y)$, together with the reflectance image, $\mathbf{I}_{unshadow}(x, y)$, are called the intrinsic images [21]. Generally, the observed image, $\mathbf{I}_{shadow}(x, y)$, can be modelled as the product of these two

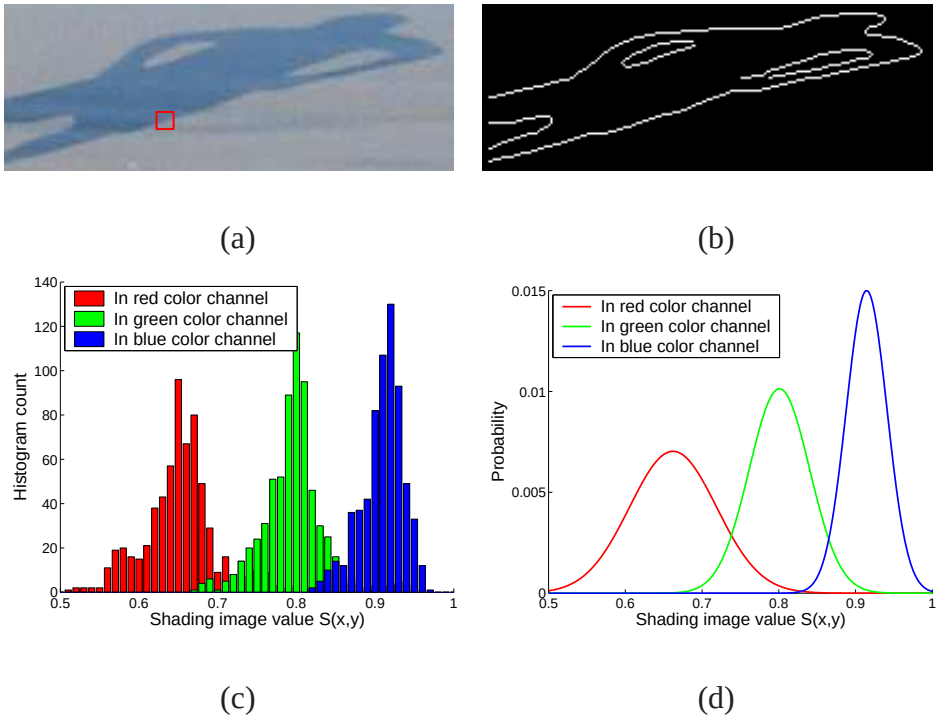


Figure 8.7: (a) A shadow patch in a target scene shown in Figure 8.11 (b). (b) The shadow boundary detected by Canny edge detector. (c) The histograms and (d) the fitted Gaussian distributions of the computed shading image values, for all pixels along the shadow boundary.

intrinsic images:

$$\mathbf{I}_{shadow}(x,y) = \mathbf{S}(x,y)\mathbf{I}_{unshadow}(x,y). \tag{8.3}$$

Therefore, our problem reduces to recovering the shading image, $\mathbf{S}(x,y)$, from the input image $\mathbf{I}_{shadow}(x,y)$.

Recently, many approaches [183, 165, 53] have been proposed to derive illumination image and reflectance image from either a single image or a video of an object under different illumination conditions. Theoretically, those decomposed light maps could be used as shadow mattes in our

work. However, the method [183] involves recording a sequence of images of a fixed outdoor scene over the course of a day, while the strategy used in [165] requires a trained classifier, which must incorporate the knowledge about the structure of the surface in the target scene and how it appears when illuminated. Therefore, these are not practical for us because in our case the target scene and its structure are not accessible. On the other hand, the more recent method [53] aims to recover intrinsic images by entropy minimization from a single image. But it is based on assumptions such as narrow-band (or spectrally-sharpened) sensors and Planckian lights, and is unable to handle the compression effects caused for instance by JPEG. In contrast, our challenge is compositing objects into a given image or a frame of a video, which are typically compressed.

Our approach takes advantage of the property that changes in color between pixels indicate either reflectance changes or shading effects [165]. In other words, it is unlikely that significant shading boundaries and reflectance edges occur at the same point. Therefore, we make the assumption that every color change along the shadow boundaries, the edges caused by illumination difference (e.g. Figure 8.7 (b)), is caused by shading only, i.e. the reflectance image colors across the shading boundaries should be the same or similar. In practice, considering the gradual change along the normal direction of the shadow boundaries, due to either compression effects or soft shadows, the input image pixel value, $\mathbf{I}_{shadow}(x, y)$, and the reflectance image pixel value, $\mathbf{I}_{unshadow}(x, y)$, of boundary pixel, (x, y), are obtained as

$$\mathbf{I}_{shadow} = median\{\mathbf{I}_{shadow}(m, n) : (m, n) \in \mathcal{N}_i\}, \tag{8.4}$$

$$\mathbf{I}_{unshadow} = median\{\mathbf{I}_{shadow}(m, n) : (m, n) \in \mathcal{N}_o\}, \tag{8.5}$$

where $\mathcal{N}_i$ and $\mathcal{N}_o$ are subsets of the set of neighbor pixels of (x, y), and the subscripts denote whether the pixels are inside ($\mathcal{N}_i$) or outside ($\mathcal{N}_o$) of the shadows. Previous methods also use color differences on two sides of a shadow boundary for estimating the influence of an illumination source [159, 99]. In our implementation, we use a 7×7 neighborhood, e.g. the pixels bounded by the red box in Figure 8.7 (a). From Equation (8.4), we can compute the shading image value as, $\mathbf{S}(x, y) = \mathbf{I}_{shadow}(x, y)/\mathbf{I}_{unshadow}(x, y)$, for each pixel (x, y) along the shadow boundaries.

Notice that for each color channel (red, green and blue), $\mathbf{S}$ is computed independently. For example, the computed $\mathbf{S}(x, y)$ along the boundaries of the shadow map (Figure 8.7 (b)) is plotted in Figure 8.7 (c,d). The interesting observation is that the changes in a color image due to shading affect the three color channels disproportionately. Evidently, the shading affects the red color channel the most and the blue color channel the least. This coincides with the observations in [121] that shadow pixels appear more "blueish". While there are some richer models to model this effect, we simply use different shading image values along three color channels to approximate the effect, i.e. $\mathbf{S} = diag(\beta_R,\ \beta_G,\ \beta_B)$. The $\mathbf{S}$ matrix is assumed approximately constant, and computed using the median value of all computed $\mathbf{S}(x, y)$ for pixels (x, y) along the shadow boundary. For each computed shadow pixel, (u, v), of the inserted object (e.g. white pixels in Figure 8.5 (b)), we are able to compute its pixel value after shading as

$$\mathbf{I}_{shadow}(u, v) = diag(\beta_R,\ \beta_G,\ \beta_B)\mathbf{I}_{unshadow}(u, v).$$

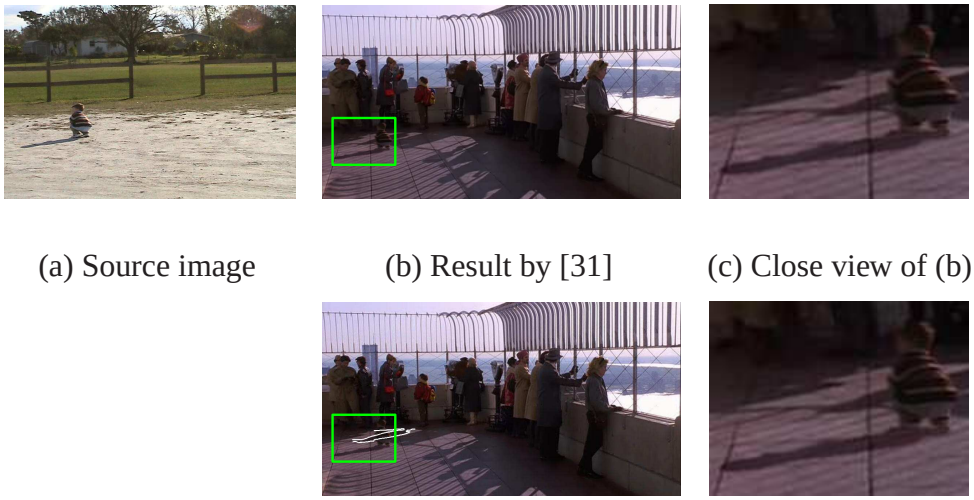
In the scene in Figure 8.7, the computed shading image values are $\beta_R = 0.67$, $\beta_G = 0.78$ and $\beta_B = 0.91$. Provided that the ground surface is locally flat and partially under shadow, which is mostly true in the real world, our experiments show that this approximation works well.

8.2.3 Results of Shadow Synthesis

The proposed method has been tested on an extensive set of target scenes. The shown target scenes vary from frames in commercial movies, frames in videos available on internet, snapshots in 3D video games to images taken by the author. In Section 8.2.3.1, we apply our method to two target images which can be calibrated. Then, in Section 8.2.3.2, we demonstrate the performance of our method on target images, where strong geometric constraints are not available. Finally, we show the applicability of our method to film production in Section 8.2.3.4.

8.2.3.1 *Scenes Where Strong Geometric Constraints are Available*

The first target frame is from the commercial movie "Sleepless in Seattle" (1993), as shown in Figure 8.2. We first compute the positions and orientations of the camera and the light source using the method described in Section 8.2.1.1. Then, we use the shadow edges marked by white lines shown in Figure 8.8 (d) to obtain the shading image values ($\beta_R = 0.48$, $\beta_G = 0.43$ and $\beta_B = 0.46$). The color characteristics of our synthesized shadow in Figure 8.8 (d) and zoomed in (e) is comparable to that by [31] in (b) and (c). However, our result is obtained by using a single frame, while their method involves 512 frames, from which we find the darkest and brightest value at each pixel as shown in Figure 8.9. In addition, it is difficult for [31] to ensure that the relationship between the light source, the reference plane, and the camera match the target due to the potential perspective distortions. Note that we multiply the R, G and B values of each pixel of



(a) Source image (b) Result by [31] (c) Close view of (b)

(d) Our result (e) Close view of (d)

Figure 8.8: The comparison of our method with the shadow matting method proposed by [31]. We aim to composite the child in the video sequence (a) into a frame sequence from the commercial movie "Sleepless in Seattle" (1993), as shown in Figure 8.2.

the composited foreground object (i.e. the child) by manually specified constant scales (0.50, 0.38 and 0.38) to match the intensity differences between the source and target image and the "reddish" effects in the target scene.

In the second experiments, we aim to paste a small statue (Figure 8.10 (a)) into the target image (Figure 8.3), based on the computed relative position and orientation with its principal axis coinciding with the computed lighting direction. The result shown in Figure 8.10 (c) demonstrates that our method is able to synthesize shadows with correct geometric relationship and realistic color characteristics. We use the shadow edges marked by red lines shown in Figure 8.10 (c) to obtain the shading image values ($\beta_R = 0.34$, $\beta_G = 0.42$ and $\beta_B = 0.51$). Note that we multiply

(a) Target shadow image (b) Target lit image

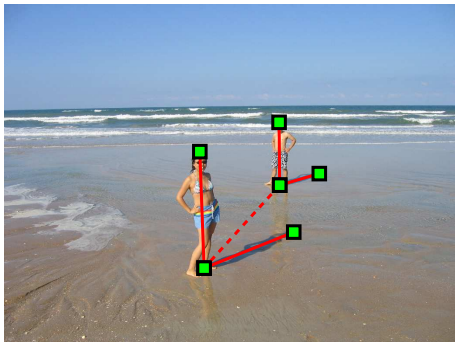
Figure 8.9: The lit and shadow images of the *Seattle sequence*. Note that we are only interested in areas where the inserted foreground might cast shadows, which in our case is located on the left bottom separated by the yellow lines.

(a) (b) (c)

Figure 8.10: We paste a small statue (a) into the target shown in Figure 8.3: (a) The image from the camera's view point; (b) The image taken by a camera whose principal axis coincides with the computed lighting direction; (c) shows the final composite with convincing shadows obtained by our new compositing method.

the intensity of each pixel of the composited foreground object, the small statue, by constant scales 1.28 to match the intensity differences between the source and the target image.

(a) Source image



(b) Target image

(c)

(d) Our result.

Figure 8.11: (c) The result by assuming that any changes in a color image due to shading should affect all three color channels proportionally.

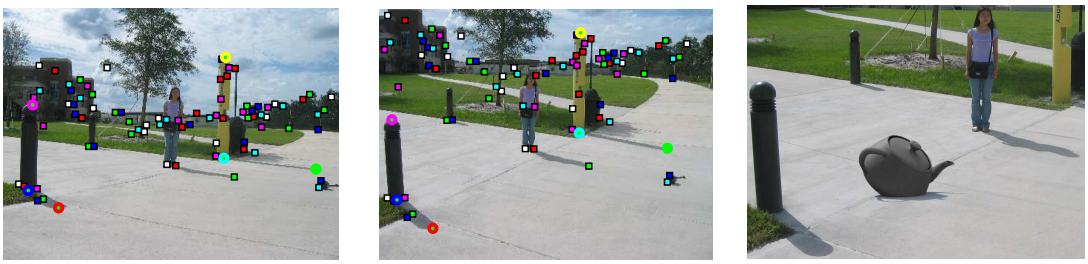
8.2.3.2 *Scenes Where Only Weak Geometric Constraints are Available*

We also created shadows for planar objects, e.g. a parking sign post as shown in Figure 8.11. The input feature points, including three pairs of points that are corresponding under a planar homology and one point used for the computation of the vertical vanishing point, are plotted in the same color in Figure 8.11 (b) as those in Figure 8.5 (a). In this experiment, we demonstrate the advantage of using different shading image values along each color channel. Our final result shown in Figure

Figure 8.12: as shown in Figure 8.5. Left: the source image. Right: a zoomed view of our result.

8.11 (d) is noticeably more realistic than the result in Figure 8.11 (c) since shadow regions are illuminated by the sky, and sky is assumed to be blue and the only source of illumination on shadowed regions [121]. The shading image for Figure 8.11 (d) is computed in Section 8.2.2, while for the result Figure 8.11 (c) we use the intensity image and compute the shading image value as $\beta_R = \beta_G = \beta_B = 0.72$.

For all of the above experiments, the light source is the sun. We also demonstrate our method for the synthetic light source from the snapshot of a 3D video game, illustrated in Figure 8.5. Whether the light source is finite or not, our method is able to synthesize the correct and realistic shadow for a inserted real player, shown in Figure 8.12. Note that the very impressive shadows are generated even for the raised hand of the real player. The weak geometric constraint is described in section 8.2.1.2. We use the shadow edges marked by red lines shown in Figure 8.12 right to obtain the shading image values ($\beta_R = \beta_G = \beta_B = 0.50$).



(a) (b) (c)

(d) (e) (f)

Figure 8.13: Image-based rendering Applications. Starting from two views (a) and (b), we first calibrate the camera and compute the light source orientation. The square marks are the corresponding points between the last two images, which are computed by using method [195]. As a result, we can render a virtual teapot with known 3D model into the real scene (b) shown in (c). Utilizing this computed geometric information, we can also insert another person (d) into (b) as shown in (f). (e) is the view from light source and is used for shadow map.

8.2.3.3 *Application to Image-based Rendering*

To demonstrate the strength and applicability of the proposed algorithm, we show two examples for augmented reality by making use of the camera and light source orientation information computed by our method. Given two views as shown in Figure 8.13 (a) and (b), the computed camera intrinsic

matrix $\mathbf{K}$ is (see Chapter 4)

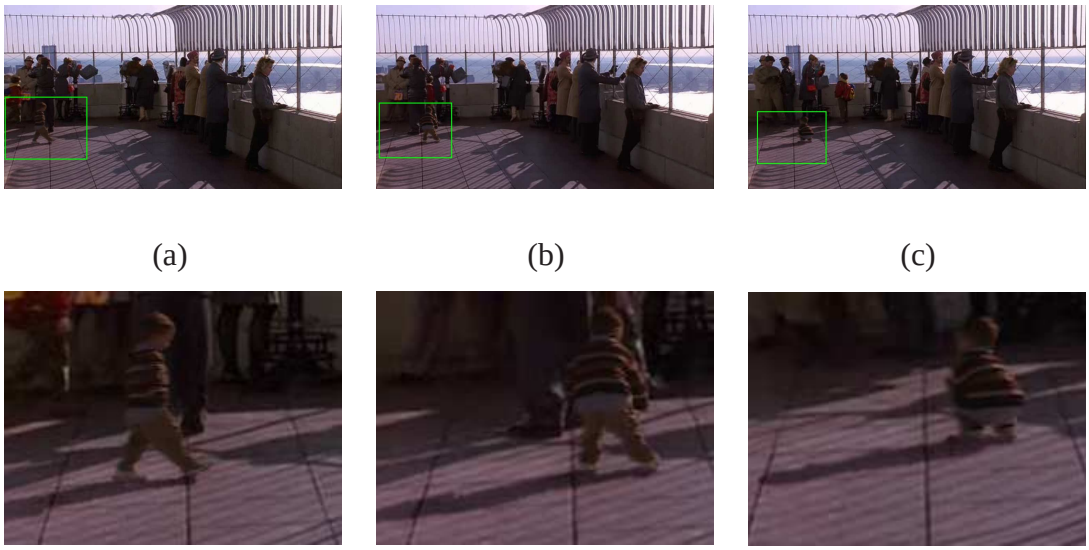
$$\mathbf{K} = \begin{bmatrix} 2641.11 & 0 & 991.70 \\ 0 & 2783.97 & 642.28 \\ 0 & 0 & 1 \end{bmatrix}, \tag{8.6}$$

and the computed polar angle ϕ and azimuthal angle θ for the light source are 47.99 and 54.91 degrees respectively. As a result, we can render a virtual teapot with known 3D model into the real scene shown in Figure 8.13 (b) and (c). The color characteristic is estimated using methods presented in Section 8.2.2. Alternatively, we can also composite the standing person extracted from Figure 8.13 (d) into Figure 8.13 (b) and synthesize its shadow using the contour of the person in Figure 8.13 (e).

8.2.3.4 *Application to Film Production*

To demonstrate the strength of our method, we apply it on two commercial Hollywood movies: "Sleepless in Seattle" (Figure 8.14) and "The Pianist" (Figure 8.15). The camera and light source parameters of the "Sleepless in Seattle" are recovered as described in Section 8.2.1.1. For the "The Pianist", we first calibrate the camera using the feature lines shown in Figure 8.16 (a). To recover the light source geometry, we manually choose twelve correspondences for the points $\mathbf{t}$, $\mathbf{s}$ and $\mathbf{b}$ as shown in Figure 8.16 (b) and (c), in the first twelve frames of the original clip.

Based on the analysis of the target frames, we observe that the light source of the "Seattle sequence" is a typical daytime sunlit, and the camera is at a location above the ground plane at

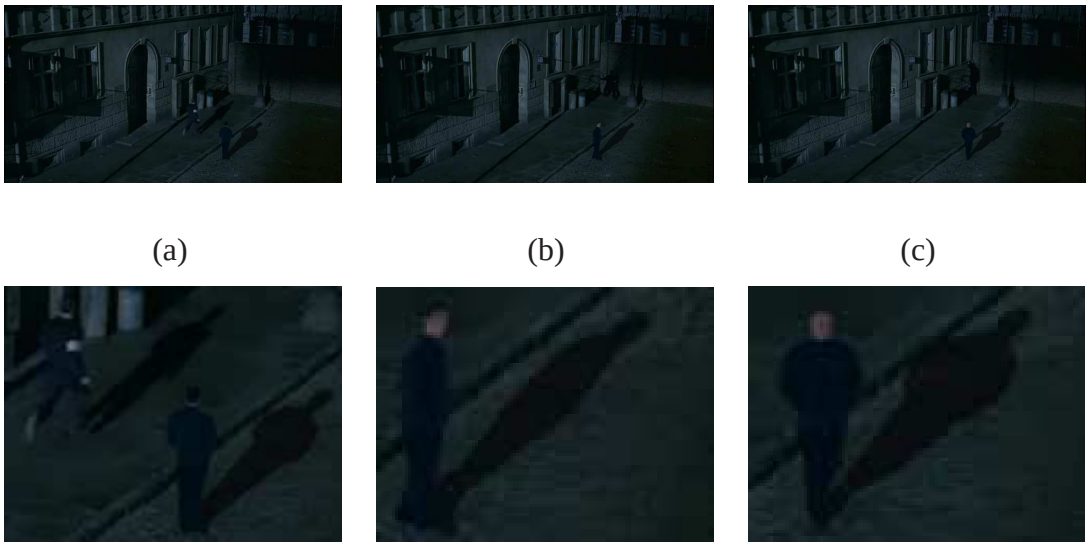


(a) (b) (c)

(d) (e) (f)

Figure 8.14: Three example frames (a), (b) and (c) of our composite of the *Seattle sequence*. The bottom row plots the zoomed in views of the frames above them.

a height of a typical person. We captured the source video at some park with similar lighting condition using an off-the-shelf Sony video camera. For the "Pianist sequence", it is very difficult to place a light at the computed light source position and then transfer the shadows extracted from the camera view to the target video, since, to do this, one needs a very dark and huge studio of about 100 meters length according to our computation to ensure the geometry matches the target background. In our case, alternatively, we place two video cameras on the 3^{rd} and 2^{nd} floors of a parking garage to capture the videos that would be seen from the original camera and light source. In other words, the shadow map [38] in 3D graphics is implemented using a real camera and applied in film production.

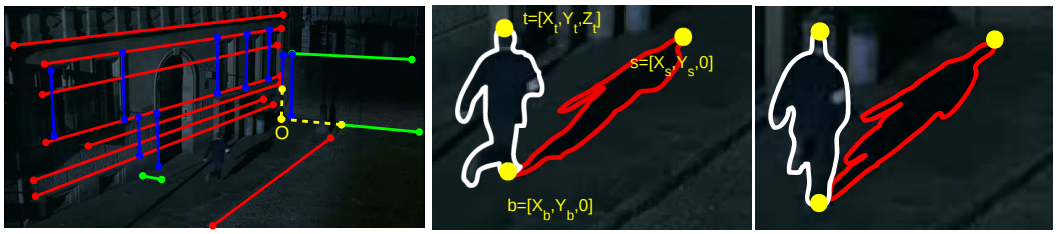


(a) (b) (c)

(d) (e) (f)

Figure 8.15: Three example frames (a), (b) and (c) of our composite of *The Pianist*. The bottom row plots the zoomed in views of the frames above them.

In both clips, our method successfully composites not only the foreground objects but also its geometrically correct and visually convincing shadows into the commercial movies without special setup. The results are shown in Figure 8.14 and Figure 8.15. In the case where the shadows of the added sequence overlap with the shadows in the original sequence, we retain the original appearance. The existing shadow areas can be computed based on the comparison of the input frame and the the target shadow image shown in Figure 8.9 (a).



(a) (b) (c)

Figure 8.16: (a) is one frame from the movie *The Pianist*, with the extracted feature lines for camera parameters estimation plotted in three different colors with the same definitions as that in Figure 8.2 (a). The two yellow dashed lines intersect at a point, O, which is specified as the image of the world coordinate origin. (b) & (c) are two close views of two frames with extracted features ($\mathbf{t}$, $\mathbf{b}$ and $\mathbf{s}$) used to compute the light source geometry.

8.3 Reflection Synthesis

For synthesizing reflection, we assume that the reflecting surface is approximately planar, and the perturbations and turbulences (e.g. waves on water) are relatively small. Two typical example scenes are shown in Figure 8.17. The laws of plane (flat surface) reflections are of course simple and well-known. However, we are dealing with an inverse problem here, since the data is only available in the 2D space through images under perspective distortion. We have two options: either we use the geometry recovered as described earlier to determine the reflections in 3D and then render images from the viewpoint of the original camera, or if possible avoid backprojection and work directly in the 2D image space. The former is rather hard and could not provide convincing results unless the geometry and the projections are almost completely error-free.

To get better and convincing results, we show that it is possible to directly obtain the correct reflecting geometry under perspective distorion of the camera. The key idea is that the reflection is governed by lines perpendicular to the reflecting surface. Therefore, points in the object and its reflection lie on a single line perpendicular to the reflecting plane, and the reflected object appears at the same distance behind the plane as the actual object is in front of it. As a result, in the 3D space a point t_i and its reflection r_i have the same perpendicular distances from the reflecting plane. However, in a perspectivly distorted image, the object and its reflection will have an identical height only if the direction of view is nearly parallel to the reflecting surface. In the example above (Fig. 8.17 (a)), however, the reflections of the standing persons are clearly viewed at a downward angle, which will make them appear foreshortened in comparison to the actual persons.

The key idea that allows us to solve the geometry of the light reflection directly in the image plane is the fact that the following cross ratio is invariant and preserved under camera perspective projection:

$$\{v_z, t_i; b_i, r_i\} = \frac{(v_z - t_i)(b_i - r_i)}{(v_z - r_i)(t_i - b_i)} = 1, \tag{8.7}$$

where v_z as before is the vertical vanishing point, and b_i is the intersection of line $t_i r_i$ and the reflection surface.

Note that the line segments $t_i b_i$ and $b_i r_i$ in the image plane have same distance only when v_z goes to infinity, e.g. the direction of view is parallel to the reflecting plane. Therefore the direct image-domain solution that we propose is straightforward and is as follows. We first compute the vertical vanishing point by identify two pairs of t_i and r_i or three points t_i, b_i and r_i. Then

(a) (b)

Figure 8.17: Two real scenes containing reflection effects.

we enforce the cross ratio property to find the correct positions of the reflections of each inserted object.

While the geometrical constraints help us to put reflections at correct positions, similar to shadows we also need to match the color characteristics of the reflections to those of the real scene. The light reflected on the object surface can be approximated as a linear combination of two reflection components: diffuse reflection component, I_d, and specular reflection component, I_s. Assuming that the camera is relatively far compared to the size of the objects being reflected, the differences between the light directions to different positions on the object are small, and hence we can use constant Fresnel factor, λ_F, for blending between diffuse I_d and specular I_s, similar to [68],

$$I_o(x, y) = \lambda_F \cdot I_s(x, y) + (1 - \lambda_F) I_d(x, y), \tag{8.8}$$

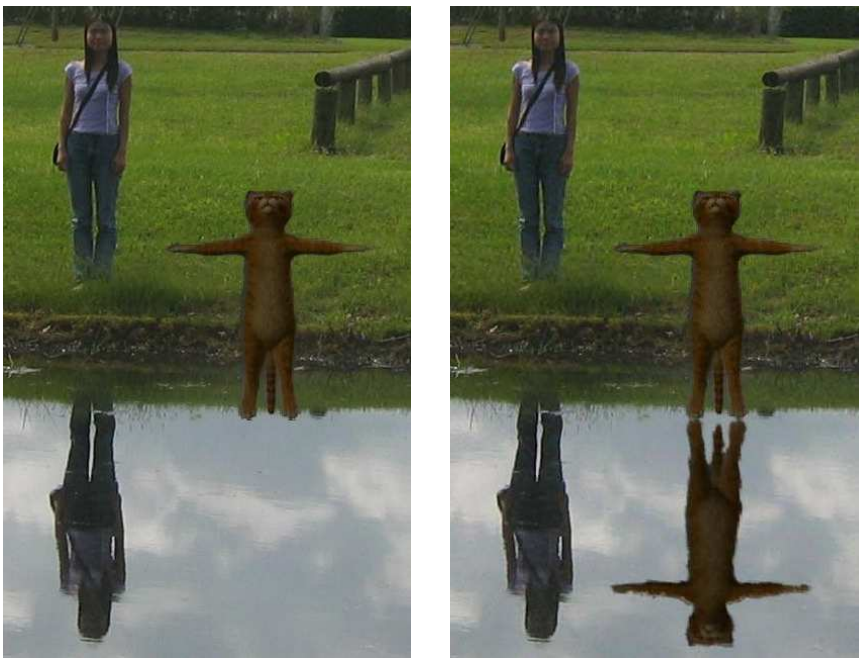
where I_o is the radiance received by the camera.

(a) (b)

Figure 8.18: Reflection synthesis of the source and target image in Figure 8.11. (a): A zoomed view of a composite with only shadow synthesis. The result (a) is noticeably fake. (b): A composite with geometrically correct and visually realistic reflections obtained by the proposed method.

The Fresnel factor can be estimated in a similar manner as the shadow opacity factor by building the histogram of existing reflection regions for each color channel and used for reflection synthesis.

To illustrate the proposed method, we tried it on different cases. In the example shown in Figure 8.18, we insert a parking sign post into a target image of a beach scene. First, we use the method described in Section 8.2 to create shadows for planar parking sign post as shown in the middel picture. However, in the target scene, we also need to synthesize the reflection effects to



(a) (b)

Figure 8.19: Synthesizing the reflections. (a) is without reflection and (b) is with reflection.

increase the realism. The bottom picture shows the composite with correct and realistic shadows and reflections.

The target scene in Figure 8.19, different from the sparse reflection in Figure 8.18, introduces challenges both in estimating the Fresnel terms and in dealing with the distortion of the water surface. Our method synthesized promising reflections of the inserted *Puss in Boots* (a character in movie Shrek 2) as shown in (b) and zoomed in (c). In the result (d) using only α blending, is the cat touching on the water surface or floating in the air?

8.4 Discussions

In this chapter, we presented methods to synthesize shadows and reflections for objects composited into target views. These methods are especially useful when the target scene is not accessible, e.g a given image or video clips from a commercial movie. We explore both the geometric and photometric constraints, and utilize them for correct and realistic shadow and reflection synthesis. The experimental results demonstrate that this method is efficient and can be applied to a variety of target scenes.

8.4.1 Shadow Synthesis

Compared to the shadow matting methods [30], our method is able to handle cases where one aims to transfer objects into inaccessible scenes and requires only a single view. In addition, our framework has advantages over the traditional "faux shadow" [187] in that it increases the geometric accuracy and visual reality of the synthesized shadows. As a pragmatic and flexible framework, it is also simple and easy to implement.

Our shadow synthesis method has a number of restrictions. First, we approximate the cast shadow texture by using the concepts of the intrinsic image and assume one dominant, point-like light source which does not model potentially complex effects arising from inter-reflections. A relatively more general shadow texture model considering penumbra effects [88, 123] can be easily apply to our framework. Second, we assume the target background to be planar. Although

there exist solutions using shadow scans [30] or weak structured lighting [18], all of those methods require hardware and nontrivial human interactions. Third, we did not present the noise existing in natural shadows in our artificially created ones. The noise can be simply modeled as a Gaussian noise to obtain more realistic visual results.

Despite these restrictions, we believe that in many settings the shadows created by our approach are more plausible than the manually generated "faux shadow". Traditional shadow generating methods typically share our requirement for a single matched key light source and have difficulties to model complex effects arising from inter-reflections. However, they can not synthesize a geometrically correct silhouette of the shadow when the view from the lighting direction is too far from the camera's viewpoint.

8.4.2 Reflection Synthesis

Synthesizing reflections for objects transferred from one natural scene into a novel scene with different lighting condition remains a hard problem. In this chapter, we focused on a slightly easier situation, where the target scene has some up-right vertical objects. We showed that this assumption leads to a novel, simple algorithm for reflection synthesis, and showed encouraging results for different target scenes. Using an image-based approach, we are unlikely to estimate realism by using known geometry of the light, the object to be pasted, and the background objects.

However, the proposed method advances the image based techniques one step further to improve the realism in applications of matting and compositing techniques.

There are a number of ways in which our current approach can be extended. First we would like to relax the vertical object constraints for the target scene. Second, we currently use the tools in Photoshop to generate the ripple effects, and would like to reduce the interaction in the future work. Finally, we would also like to utilize richer models in learning the illumination conditions from the target scene and apply it to the composited objects.

CHAPTER 9

CONCLUSION

In this dissertation, we successfully showed that geometrically correct and visually realistic video post-production is possible even among videos captured by unknown moving cameras.

We proposed two novel constraints for camera calibration: first, the inter-frame constraints extend the current state-of-the-art to situations where only one vanishing point is available. Second, the constraint making use of constant motion is especially useful in applications such as self-calibration from turn-table sequences and calibration of a camera network, where the cameras are arranged in angularly equal positions on a circle. We developed a new framework for video analysis and post-production among videos captured by cameras undergoing distinct general or special motions. This framework is able to enforce the geometrical correctness and photometric consistency.

This work addresses the following problems in different areas: camera calibration, camera motion analysis, single view geometry, shadow synthesis, video alignment, video segmentation, alpha matting and image-based rendering. However, all of these problems are closely related and

demonstrate the important fact that video post-production techniques will be useful and attractive if we successfully resolve the related vision problems.

LIST OF REFERENCES

[1] M. Antone and M. Bosse. "Calibration of outdoor cameras from cast shadows." In *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, pp. 3040–3045, 2004.

[2] M. Agrawal and L. Davis. "Complete camera calibration using spheres: Dual space approach." In *Proc. IEEE ICCV*, pp. 782–789, 2003.

[3] N. E. Apostoloff and A.W. Fitzgibbon. "Bayesian video matting using learnt image priors." In *Proc. IEEE CVPR*, pp. 407–414, 2004.

[4] L. De Agapito, E. Hayman, and I. Reid. "Self-calibration of rotating and zooming cameras." *Int. J. Comput. Vision*, **45**(2):107–127, 2001.

[5] S. Ayer and H. Sawhney. "Layered Representation of Motion Video Using Robust Maximum-Likelihood Estimation of Mixture Models and Mdl Encoding." In *Proc. IEEE ICCV*, 1995.

[6] A. Agarwala, D. Salesin, and S. Seitz. "Keyframe-based tracking for rotoscoping and animation." *ACM Trans. Graph.*, **23**(3):584–591, 2004.

[7] G. Medioni A. Francois and R. Waupotitsch. "Reconstructing mirror symmetric scenes from a single view using 2-view stereo geometry." In *Proc. ICPR*, pp. 12–16, 2002.

[8] M. Armstrong, A. Zisserman, and R.I. Hartley. "Self-Calibration from Image Triplets." In *Proc. ECCV*, pp. 3–16, 1996.

[9] C. Buchler, M. Bosse, S. Gortler L. McMillan, and M. Cohen. "Unstructured Lumigraph Rendering." In *Proc. ACM SIGGRAPH*, pp. 425–432, 2001.

[10] M. Bertalmio, A. Bertozzi, and G. Sapiro. "Navier-stokes, fluid dynamics, and image and video inpainting." In *Proc. IEEE CVPR*, pp. 355–362, 2001.

[11] M. Ben-Ezra. "Segmentation with invisible keying signal." In *Proc. IEEE CVPR*, pp. 32–37, 2000.

[12] S. Boivin and A. Gagalowicz. "Image-based rendering of diffuse, specular and glossy surfaces from a single image." In *Proc. ACM SIGGRAPH*, pp. 107–116, 2001.

[13] P. Bouthemy, M. Gelgon, and F. Ganansia. “A unified approach to shot change detection and camera motion characterization.” *IEEE Trans. Circuits Syst. Video Technol.*, **9**(7):1030–1044, 1999.

[14] A. Bjorck. *Numerical Methodes for Least Squares Problems*. SIAM, Philadelphia, 1996.

[15] L. Bergen and F. Meyer. “A Novel Approach to Depth Ordering in Monocular Image Sequences.” In *Proc. IEEE CVPR*, pp. 536–541, 2000.

[16] S. Bougnoux. “From Projective to Euclidean Space Under any Practical Situation, a Criticism of Self-Calibration.” In *Proc. IEEE ICCV*, pp. 790–796, 1998.

[17] E. Boyer. “Object Models From Contour Sequences.” In *Proc. ECCV*, pp. 109–118, 1996.

[18] J. Bouguet and P. Perona. “3D Photography on Your Desk.” In *Proc. IEEE ICCV*, pp. 43–50. IEEE Computer Society, 1998.

[19] R. Brinkman. *The Art and Science of Digital Compositing*. Morgan Kaufman, 1999.

[20] A. Bryson. *Dynamic Optimization*. Addison Wesley, 1999.

[21] H.G. Barrow and J.M. Tenenbaum. *Recovering intrinsic scene characteristics from images*. Academic Press, 1978.

[22] Y. Chuang, A. Agarwala, B. Curless, D. Salesin, and R. Szeliski. “Video Matting of Complex Scenes.” *ACM Trans. Graph.*, **21**(3):243–248, 2002.

[23] C. Colombo, A.D. Bimbo, and F. Pernici. “Metric 3D Reconstruction and Texture Acquisition of Surfaces of Revolution from a Single Uncalibrated View.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **27**(1):99–114, 2005.

[24] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski. “A Bayesian Approach to Digital Matting.” In *Proc. IEEE CVPR*, volume 2, pp. 264–271, 2001.

[25] R. Cipolla, T. Drummond, and D. Robertson. “Camera calibration from vanishing points in images of architectural scenes.” In *Proc. of BMVC*, pp. 382–391, 1999.

[26] X. Cao and H. Foroosh. “Camera Calibration Without Metric Information Using 1D Objects.” In *Proc. IEEE ICIP*, pp. 1349–1352, 2004.

[27] X. Cao and H. Foroosh. “Easy Camera Calibration From Inter-Image Homographies.” In *Proc. of IEEE International Workshop on Image and Video Registration in CVPR*, 2004.

[28] X. Cao and H. Foroosh. “Metrology from Vertical Objects.” In *Proc. of BMVC*, 2004.

[29] X. Cao and H. Foroosh. "Simple Calibration Without Metric Information Using an Isosceles Trapezoid." In *Proc. ICPR*, pp. 104–107, 2004.

[30] Y. Chuang, D. Goldman, B. Curless, D. Salesin, and R. Szeliski. "Shadow Matting and Compositing." *ACM Trans. Graph.*, **22**(3):494–500, 2003.

[31] Y. Chuang. *New Models and Methods for Matting and Compositing*. PhD thesis, University of Washington, 2004.

[32] Y. Caspi and M. Irani. "A step towards sequence to sequence alignment." In *Proc. IEEE CVPR*, pp. 682–689, 2000.

[33] Y. Caspi and M. Irani. "Alignment of Non-Overlapping sequences." In *Proc. IEEE ICCV*, pp. 76–83, 2001.

[34] R. Carceroni, F. Padua, G. Santos, and K. Kutulakos. "Linear Sequence-to-Sequence Alignment." In *Proc. IEEE CVPR*, pp. 746–753, 2004.

[35] A. Criminisi, P. Perez, and K. Toyama. "Region filling and object removal by exemplar-based inpainting." *IEEE Trans. Image Process*, **13**(9):1200–1212, 2004.

[36] S. Chaudhuri and A. N. Rajagopalan. *Depth from Defocus: A Real Aperture Imaging Approach*. Springer-Verlag, 1999.

[37] A. Criminisi. *Accurate Visual Metrology From Single and Multiple Uncalibrated Images*. Springer-Verlag, 2001.

[38] F. C. Crow. "Shadow algorithms for computer graphics." In *Proc. of SIGGRAPH*, pp. 242–248, 1977.

[39] A. Criminisi, I. Reid, and A. Zisserman. "Single View Metrology." *Int. J. Comput. Vision*, **40**(2):123–148, 2000.

[40] X. Cao and M. Shah. "Camera Calibration and Light Source Estimation from Images with Shadows." In *Proc. IEEE CVPR*, pp. 918–923, 2005.

[41] X. Cao and M. Shah. "Creating Realistic Shadows of Composited Objects." In *Proc. WACV*, pp. 294–299, 2005.

[42] Y. Caspi, D. Simakov, and M. Irani. "Feature-based sequence-to-sequence matching." In *Proc. VAMODS workshop with ECCV*, 2002.

[43] X. Cao, Y. Shen, M. Shah, and H. Foroosh. "Single View Compositing with Shadows." *The Visual Computer*, **21**(8):639–648, 2005.

[44] B. Caprile and V. Torre. "Using Vanishing Points for Camera Calibration." *Int. J. Comput. Vision*, **4**(2):127–140, 1990.

[45] Q. Chen, H. Wu, and T. Wada. "Camera Calibration with Two Arbitrary Coplanar Circles." In *Proc. ECCV*, pp. 521–532, 2004.

[46] P. Debevec, A. Wenger, C. Tchou, A. Gardner, J. Waese, and T. Hawkins. "A lighting reproduction approach to live-action compositing." *ACM Trans. Graph.*, **21**(3):547–556, 2002.

[47] L. Duan, M. Xu, Q. Tian, and C. Xu. "Nonparametric motion model with applications to camera motion pattern classification." In *Proc. ACM MULTIMEDIA*, pp. 328–331, 2004.

[48] O. Faugeras. "What can be seen in three dimensions with an uncalibrated stereo rig?" In *Proc. ECCV*, pp. 563–578, 1992.

[49] O. Faugeras. *Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.

[50] H. Foroosh, M. Balci, and X. Cao. "Self-Calibrated Reconstruction of Partially Viewed Symmetric Objects." In *Proc. IEEE ICASSP*, pp. 869–872, 2005.

[51] H. Foroosh, X. Cao, and M. Balci. "Metrology in Uncalibrated Images Given One Vanishing Point." In *Proc. IEEE ICIP*, pp. 361–364, 2005.

[52] A. W. Fitzgibbon, G. Cross, and A. Zisserman. "Automatic 3D Model Construction for Turn-Table Sequences." In *SMILE Wkshp.*, pp. 155–170, 1998.

[53] G. Finlayson, M. Drew, and C. Lu. "Intrinsic Images by Entropy Minimization." In *Proc. ECCV*, pp. 582–595, 2004.

[54] J.M. Frahm and R. Koch. "Camera Calibration with Known Rotation." In *Proc. IEEE ICCV*, pp. 1418–1425, 2003.

[55] O. Faugeras and Q.T. Luong. *The Geometry of Multiple Images*. MIT Press, 2001.

[56] O. Faugeras, T. Luong, and S. Maybank. "Camera self-calibration: theory and experiments." In *Proc. of ECCV*, pp. 321–334, 1992.

[57] A. Fitzgibbon, Y. Wexler, and A. Zisserman. "Image-Based Rendering Using Image-Based Priors." In *Proceedings of IEEE ICCV*, 2003.

[58] P. Gurdjos, A. Crouzil, and R. Payrissat. "Another Way of Looking at Plane-Based Calibration: the Centre Circle Constraint." In *Proc. ECCV*, pp. 252–266, 2002.

[59] P. Giaccone and G. Jones. "Segmentation of Global Motion using Temporal Probabilistic Classification." In *Proc. of BMVC*, 1998.

[60] M. A. Giese and T. Poggio. “Synthesis and Recognition of Biological Motion Patterns Based on Linear Superposition of Prototypical Motion Sequences.” In *Proc. IEEE Workshop on Multi-View Modeling & Analysis of Visual Scenes*, p. 73, 1999.

[61] P. Gurdjos and P. Sturm. “Methods and Geometry for Plane-Based Self-Calibration.” In *Proc. IEEE CVPR*, pp. 491–496, 2003.

[62] A. Heyden and K. Astrom. “Euclidean reconstruction from image sequences with varying and unknown focal length and principal point.” In *Proc. IEEE CVPR*, pp. 438–443, 1997.

[63] A. Heyden and K. Astrom. “Flexible Calibration: Minimal Cases for Auto-Calibration.” In *Proc. IEEE ICCV*, pp. 350–355, 1999.

[64] http://www.robots.ox.ac.uk/˜az/HZbook/HZfigures.html.

[65] R. I. Hartley. “Estimation of Relative Camera Positions for Uncalibrated Cameras.” In *Proc. ECCV*, pp. 579–587, 1992.

[66] R. I. Hartley. “Self-Calibration of Stationary Cameras.” *Int. J. Comput. Vision*, **22**(1):5–23, 1997.

[67] R.I. Hartley. “Theory and practice of projective rectification.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **35**(2):115–127, 1999.

[68] W. Heidrich. *High-quality Shading and Lighting for Hardware-accelerated Rendering*. PhD thesis, Univ. of Erlangen, 1999.

[69] J. Heikkila. “Geometric camera calibration using circular control points.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**(10):1066–1077, 2000.

[70] T. S. Huang and O. Faugeras. “Some properties of the e matrix in twoview motion estimation.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **11**(12):1310–1312, 1989.

[71] T. Horprasert, D. Harwood, and L.S. Davis. “A Statistical Approach for Real-Time Robust Background Subtraction and Shadow Detection.” In *Proc. IEEE ICCV, Frame Rate Workshop*, pp. 1–19, 1999.

[72] P. Hillman, J. Hannah, and D. Renshaw. “Alpha Channel Estimation in High Resolution Images and Image Sequences.” In *Proc. IEEE CVPR*, pp. 1063–1068, 2001.

[73] B. K. P. Horn. *Robot Vision*. McGraw-Hill, 1986.

[74] R. I. Hartley and P. Sturm. “Triangulation.” In *Proc. CAIP*, pp. 190–197, 1995.

[75] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[76] Y.T. Jia, S.M. Hu, and R.R. Martin. "Video completion using tracking and fragment merging." *The Visual Computer*, **21**(8):601–610, 2005.

[77] R. Jin, A. Hauptmann, and Y. Qi. "A Probabilistic Model for Camera Zoom Motion Detection." In *Proc. ICPR*, pp. 859–862, 2002.

[78] B. Johansson. *Computer Vision Using Rich Features - Geometry and Systems*. PhD thesis, Dept. of Mathematics, Lund University, 2002.

[79] G. Jiang, L. Quan, and H. T. Tsui. "Circular Motion Geometry Using Minimal Data." *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(6):721–731, 2004.

[80] O. Javed and M. Shah. "Tracking and Object Classification for Automated Surveillance." In *Proc. ECCV*, pp. 343–357, 2002.

[81] G. Jiang, H. T. Tsui, L. Quan, and A. Zisserman. "Single Axis Geometry by Fitting Conics." *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**(10):1343–1348, 2003.

[82] J. Jia, T. Wu, Y. Tai, and C. Tang. "Video repairing: inference of foreground and background under severe occlusion." In *Proc. IEEE CVPR*, p. 364C371, 2004.

[83] D. Kriegman and P. Belhumeur. "What Shadows Reveal about Object Structure." In *Proc. ECCV*, pp. 399–414, 1998.

[84] A. Klaus, J. Bauer, K. Karner, P. Elbischger, R. Perko, and H. Bischof. "Camera Calibration from a Single Night Sky Image." In *Proc. IEEE CVPR*, pp. 151–157, 2004.

[85] J. J. Koenderink and A. J. Van Doorn. "Photometric invariants related to solid shape." *Optica Acta*, **27**(7):981–996, 1980.

[86] Q. Ke and T. Kanade. "A Subspace Approach to Layer Extraction." In *Proc. IEEE CVPR*, 2001.

[87] Q. Ke and T. Kanade. "A Robust Subspace Approach to Layer Extraction." In *Proc. IEEE Motion*, 2002.

[88] B. Keating and N. Max. "Shadow Penumbras for Complex Objects by Depth-Dependent Filtering of Multi-Layer Depth Images." In *Proc. Eurographics Workshop on Rendering*, pp. 205–220. Springer-Verlag, 1999.

[89] N. Krahnstoever and P. R. S. Mendonça. "Bayesian Autocalibration for Surveillance." In *Proc. IEEE ICCV*, pp. 1858–1865, 2005.

[90] D. Kersten, P. Mamassian, and D. C. Knill. "Moving cast shadows induce apparent motion in depth." *Perception*, **26**(2):171–192, 1997.

[91] K.N. Kutulakos and S.M. Seitz. "A Theory of Shape by Space Carving." In *Proc. IEEE CVPR*, pp. 307–314, 1999.

[92] S. Khan and M. Shah. "Object Based Segmentation of Video Using Color, Motion and Spatial." In *Proc. IEEE CVPR*, pp. 746–751, 2001.

[93] F. Kahl, B. Triggs, and K. Åström. "Critical Motions for Auto-Calibration When Some Intrinsic Parameters Can Vary." *J. Math. Imaging Vis.*, **13**(2):131–146, 2000.

[94] S. Kang and R. S. Weiss. "Can We Calibrate a Camera Using an Image of a Flat, Textureless Lambertian Surface?" In *Proc. ECCV*, pp. 640–653, 2000.

[95] V. Kolmogorov and R. Zabih. "Multi-camera Scene Reconstruction via Graph Cut." In *Proceedings of ECCV*, 2002.

[96] I. Laptev, S. Belongie, P. Perez, and J. Wills. "Periodic Motion Detection and Segmentation via Approximate Sequence Alignment." In *Proc. IEEE ICCV*, pp. 816–823, 2005.

[97] M. Levoy and P. Hanrahan. "Light Field Rendering." In *Proc. ACM SIGGRAPH*, 1996.

[98] C. Lin, A. Huertas, and R. Nevatia. "Detection of Buildings Using Perceptual Groupings and Shadows." pp. 62–69, 1994.

[99] Y. Li, S. Lin, H. Lu, and H. Shum. "Multiple-cue Illumination Estimation in Textured Scenes." In *Proc. IEEE ICCV*, pp. 1366–1373, 2003.

[100] D. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints." *Int. J. Comput. Vision*, **60**(2):91–110, 2004.

[101] F. Liu and R. Picard. "Finding periodicity in space and time." In *Proc. IEEE ICCV*, pp. 376–383, 1998.

[102] Y. Li, J. Sun, and H. Shum. "Video Object Cut and Paste." *ACM Trans. Graph.*, **24**(3):595–600, 2005.

[103] N. Levi and M. Werman. "The viewing graph." In *Proc. IEEE CVPR*, pp. 518–524, 2003.

[104] D. Liebowitz and A. Zisserman. "Metric Rectification for Perspective Images of Planes." In *Proc. IEEE CVPR*, pp. 482–488, 1998.

[105] D. Liebowitz and A. Zisserman. "Combining Scene and Auto-Calibration Constraints." In *Proc. IEEE ICCV*, pp. 293–300, 1999.

[106] F. Lv, T. Zhao, and R. Nevatia. "Self-Calibration of a Camera from Video of a Walking Human." In *Proc. ICPR*, pp. 562–567, 2002.

[107] E. Malis and R. Cipolla. "Camera self-calibration from unknown planar structures enforcing the multi-view constraints between collineations." *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**(9):1268–1272, 2002.

[108] I. Mikic, P. Cosman, G. Kogut, and M. Trivedi. "Moving Shadow and Object Detection in Traffic Scenes." In *Proc. ICPR*, volume 1, pp. 321–324, 2000.

[109] A. Mittal and L. Davis. "M2Tracker: A Multi-view Approach to Segmenting and Tracking People in a Cluttered Scene Using Region-Based Stereo." In *Proc. ECCV*, pp. 18–36, 2002.

[110] P. R. S. Mendonça. *Multiview Geometry: Profiles and Self-Calibration*. PhD thesis, University of Cambridge, Cambridge, UK, May 2001.

[111] X. Meng and Z. Hu. "A new easy camera calibration technique based on circular points." *Pattern Recognition*, **36**(5):1155–1164, 2003.

[112] Y. Mishima. "Soft edge chroma-key generation based upon hexoctahedral color space.", 1993. U.S. Patent 5,355,174.

[113] G. F. McLean and D. Kotturi. "Vanishing point detection by line clustering." *IEEE Trans. Pattern Anal. Mach. Intell.*, **17**(11):1090–1095, 1995.

[114] J. More. *The Levenberg-Marquardt Algorithm, Implementation, and Theory*. Springer-Verlag, numerical analysis edition, 1977.

[115] W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, and L. McMillan. "Image-based 3D Photography using Opacity Hulls." In *Proc. ACM SIGGRAPH*, pp. 427–437, 2002.

[116] J. Malik and R. Rosenholtz. "Computing local surface orientation and shape from texture for curved surfaces." *Int. J. Comput. Vision*, **23**(2):149–168, 1997.

[117] H. Mitsumoto, S. Tamura, K. Okazaki, N. Kajimi, and Y. Fukui. "3-D Reconstruction Using Mirror Images Based on a Plane Symmetry Recovering Method." *IEEE Trans. Pattern Anal. Mach. Intell.*, **14**(9):941–946, 1992.

[118] P. R. S. Mendonça, K-Y. K. Wong, and R. Cipolla. "Camera pose estimation and reconstruction from image profiles under circular motion." In *Proc. ECCV*, pp. 864–877, 2000.

[119] P. R. S. Mendonça, K-Y. K. Wong, and R. Cipolla. "Epipolar Geometry from Profiles under Circular Motion." *IEEE Trans. Pattern Anal. Mach. Intell.*, **23**(6):604–616, 2001.

[120] T. Mitsunaga, T. Yokoyama, and T. Totsuka. “Autokey: Human assisted key extraction.” In *Proc. ACM SIGGRAPH*, pp. 265–272, 1995.

[121] S. Nadimi and B. Bhanu. “Physical Models for Moving Shadow and Object Detection in Video.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(8):1079–1087, 2004.

[122] C. Ngo. *Analysis of Spatio-Temporal Slices for Video Content Representation*. PhD thesis, Hong Kong University of Science & Technology, 2000.

[123] H. Nicolas. “Shadow Synthesis for Video Postproduction.” *IEEE Signal Processing Letters*, **12**(4):321–324, 2005.

[124] W. Niem. “Robust and Fast Modelling of 3D Natural Objects from Multiple Views.” In *Proc. SPIE*, volume 2182, pp. 388–397, 1994.

[125] D. Nister. “An efficient solution to the five-point relative pose problem.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(6):756–770, 2004.

[126] M. Oren and S. K. Nayar. “A theory of specular surface geometry.” *Int. J. Comput. Vision*, **24**(2):105–124, 1996.

[127] T. Okabe, I. Sato, and Y. Sato. “Spherical Harmonics vs. HaarWavelets: Basis for Recovering Illumination from Cast Shadows.” In *IEEE Conference on CVPR*, pp. 50–57, 2004.

[128] A. Ostler. “The Primal seas: water on PlayStation 2.” In *Proc. ACM SIGGRAPH on Sketches & applications*, pp. 1–1, 2003.

[129] T. Porter and T. Duff. “Compositing digital iamges.” In *Proc. ACM SIGGRAPH*, pp. 253–259, 1984.

[130] A. P. Pentland. “A new sense for depth of field.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **9**(4):523–531, 1987.

[131] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[132] L. Petrovic, B. Fujito, L. Williams, and A. Finkelstein. “Shadows for Cel Animation.” In *Proc. ACM SIGGRAPH*, pp. 511–516, 2000.

[133] P. Perez, M. Gangnet, and A. Blake. “Poisson Image Editing.” In *Proc. ACM SIGGRAPH*, pp. 313–318, 2003.

[134] M. Pollefeys, R. Koch, and L. V. Gool. “Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters.” *Int. J. Comput. Vision*, **32**(1):7–25, 1999.

[135] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara. "Detecting Moving Shadows: Algorithms and Evaluation." *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**(7):918–923, 2003.

[136] N. V. Patel and I. K. Sethi. "Video shot detection and characterization for video databases." *Pattern Recognition*, **30**(4):583–592, 1997.

[137] K. Patwardhan, G. Sapiro, and M. Bertalmio. "Video inpainting of occluding and occluded objects." In *Proc. IEEE ICIP*, 2005.

[138] L. Quan and O.D. Faugeras. "The fundamental matrix: Theory, algorithms, and stability analysis." *Int. J. Comput. Vision*, **17**(1):43–75, 1996.

[139] L. Quan and O.D. Faugeras. "Self-Calibration of a Moving Camera from Point Correspondences and Fundamental Matrices." *Int. J. Comput. Vision*, **22**(3):261–289, 1997.

[140] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood. "View-invariant Alignment and Matching of Video Sequences." In *Proc. IEEE ICCV*, pp. 939–945, 2003.

[141] C. Rother, V. Kolmogorov, and A. Blake. ""GrabCut": interactive foreground extraction using iterated graph cuts." *ACM Trans. Graph.*, **23**(3):309–314, 2004.

[142] I. Reid and A. North. "3D Trajectories from a Single Viewpoint using Shadows." In *Proc. of BMVC*, 1998.

[143] M. Ruzon and C. Tomasi. "Alpha Estimation in Natural Images." In *Proc. IEEE CVPR*, pp. 18–25, 2000.

[144] A. R. Smith and J. F. Blinn. "Blue screen matting." In *Proc. ACM SIGGRAPH*, pp. 259–268, 1996.

[145] P. Smith, T. Drummond, and R. Cipolla. "Layered Motion Segmentation and Depth Ordering by Tracking Edges." *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(4):479–494, 2004.

[146] J. Shade, S. Gortler, L. He, and R. Szeliski. "Layered Depth Images." In *Proc. ACM SIGGRAPH*, pp. 231–242, 1998.

[147] R. Swaminathan, M.D. Grossberg, and S.K. Nayar. "A Perspective on Distortions." In *Proc. IEEE CVPR*, pp. 594–601, 2003.

[148] Y. Seo and K. Hong. "About the Self-Calibration of a Rotating and Zooming Camera: Theory and Practice." In *Proc. IEEE ICCV*, pp. 183–189, 1999.

[149] J. A. Shufelt. "Performance Evaluation and Analysis of Vanishing Point Detection Techniques." *IEEE Trans. Pattern Anal. Mach. Intell.*, **21**(3):282–288, 1999.

[150] J. Sun, J. Jia, C.K. Tang, and H.Y. Shum. "Poisson matting." *ACM Trans. Graph.*, **23**(3):315–321, 2004.

[151] G. Sudhir and J. C. M. Lee. "Video annotation by motion interpretation using optical flow streams." *J. Visual Commun. Image Representation*, **4**:354–368, 1996.

[152] J. Shi and J. Malik. "Motion Segmentation and Tracking Using Normalized Cuts." In *Proc. IEEE ICCV*, 1998.

[153] P. Sturm and S. Maybank. "On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications." In *Proc. IEEE CVPR*, pp. 432–437, 1999.

[154] J. Stauder, R. Mech, and J. Ostermann. "Detection of Moving Cast Shadows for Object Segmentation." *IEEE Trans. Multimedia*, **1**(1):65–76, 1999.

[155] S. Sullivan and J. Ponce. "Automatic Model Construction, Pose Estimation, and Object Recognition from Photographs Using Triangular Splines." *IEEE Trans. Pattern Anal. Mach. Intell.*, **20**(10):1091–1097, 1998.

[156] S. Sinha, M. Pollefeys, and L. McMillan. "Camera Network Calibration from Dynamic Silhouettes." In *Proc. IEEE CVPR*, pp. 195– 202, 2005.

[157] C. E. Springer. *Geometry and Analysis of Projective Spaces*. Freeman, 1964.

[158] D. Scharstein and R. Szeliski. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms." *Int. J. Comput. Vision*, **47**(1), 2002.

[159] I. Sato, Y. Sato, and K. Ikeuchi. "Stability issues in recovering illumination distribution from brightness in shadows." In *Proc. IEEE CVPR*, pp. 400–407, 2001.

[160] I. Sato, Y. Sato, and K. Ikeuchi. "Illumination from shadows." *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**(3):290–300, 2003.

[161] P. Sand and S. Teller. "Video matching." *ACM Trans. Graph.*, **23**(3):592–599, 2004.

[162] G.P. Stein. "Tracking from multiple view points: Self-calibration of space and time." In *DARPA IU Workshop*, pp. 521–527, 1998.

[163] P. Sturm. "Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction." In *Proc. IEEE CVPR*, pp. 1100–1105, 1997.

[164] R. Szeliski. "Shape from Rotation." In *Proc. IEEE CVPR*, pp. 625–630, 1991.

[165] M. F. Tappen, W. T. Freeman, and E. H. Adelson. "Recovering Intrinsic Images from a Single Image." In *Advances in Neural Information Processing Systems*, 2002.

[166] T. Tuytelaars and L. Van Gool. "Synchronizing Video Sequences." In *Proc. IEEE CVPR*, volume 1, pp. 762–768, 2004.

[167] P. Torr and D. Murray. "Outlier Detection and Motion Segmentation." In *Proc. SPIE Sensor Fusion Conference V*, pp. 432–443, 1993.

[168] B. Triggs, P. McLauchlan, R. I. Hartley, and A. Fitzgibbon. "Bundle Adjustment — A Modern Synthesis." In *Vision Algorithms: Theory and Practice*, pp. 298–373, 1999.

[169] P. H. S. Torr. "Bayesian Model Estimation and Selection for Epipolar Geometry and Generic Manifold Fitting." *Int. J. Comput. Vision*, **50**(1):35–61, 2002.

[170] P. Tresadern and I. Reid. "Synchronizing Image Sequences of Non-Rigid Objects." In *Proc. BMVC*, pp. 629–638, 2003.

[171] H. Trivedi. "Can multiple views make up for lack of camera registration?" *Image Vision Comput.*, **6**(1):29–32, 1988.

[172] B. Triggs. "Autocalibration and the Absolute Quadric." In *Proc. IEEE CVPR*, pp. 609–614, 1997.

[173] B. Triggs. "Autocalibration from planar scenes." In *Proc. ECCV*, pp. 89–105, 1998.

[174] R.Y. Tsai. "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses." *IEEE J. of Robotics and Automation*, **3**(4):323–344, 1987.

[175] H. Tao, H. Sawhney, and R. Kumar. "Object Tracking with Bayesian Estimation of Dynamic Layer Representations." *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**(1):75–89, 2002.

[176] H. Teramoto and G. Xu. "Camera Calibration by a Single Image of Balls: From Conics to the Absolute Conic." In *Proc. ACCV*, pp. 499–506, 2002.

[177] S. Ullman. *The interpretation of visual motion*. MIT Press, Cambridge,USA, 1979.

[178] R. Vidal and Y. Ma. "A Unified Algebraic Approach to 2-D and 3-D Motion Segmentation." In *Proc. ECCV*, 2004.

[179] R. Vidal, Y. Ma, and J. Piazzi. "A New GPCA Algorithm for Clustering Subspaces by Fitting, Differentiating and Dividing Polynomials." In *Proc. IEEE CVPR*, pp. 510–517, 2004.

[180] L. Van Gool, M. Proesmans, and A. Zisserman. "Planar homologies as a basis for grouping and recognition." *Image and Vision Computing*, **16**:21–26, January 1998.

[181] J. Wang and E. Adelson. "Representing Moving Images with Layers." *IEEE Trans. on Image Processing*, **3**(5):625–638, 1994.

[182] J. Wills, S. Agarwal, and S. Belongie. "What Went Where." In *Proc. IEEE CVPR*, 2003.

[183] Y. Weiss. "Deriving intrinsic images from image sequences." In *Proc. IEEE ICCV*, pp. 68–75, 2001.

[184] F.C. Wu, Z.Y. Hu, and H.J. Zhu. "Camera calibration with moving one-dimensional objects." *Pattern Recognition*, **38**(5):755–765, 2005.

[185] A. Whitehead, R. Laganiere, and P. Bose. "Temporal Synchronization of Video Sequences in Theory and in Practice." In *Proc. IEEE WACV/MOTION*, pp. 132–137, 2005.

[186] K.-Y. Wong, R.S.P. Mendonça, and R. Cipolla. "Camera Calibration from Surfaces of Revolution." *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**(2):147–161, 2003.

[187] S. Wright. *Digital Compositing for Film and Video*. Focal Press, 2001.

[188] Y. Wexler and D. Simakov. "Space-Time Scene Manifolds." In *Proc. IEEE ICCV*, pp. 858–863, 2005.

[189] Y. Wexler, E. Shechtman, and M. Irani. "Space-time video completion." In *Proc. IEEE CVPR*, pp. 120–127, 2004.

[190] L. Wolf and A. Zomet. "Sequence-to-Sequence Self Calibration." In *Proc. ECCV*, pp. 370–382, 2002.

[191] J. Xiao, X. Cao, and H. Foroosh. "Object Transfer Between Non-Overlapping Videos." In *Proc. IEEE VR*, 2006.

[192] J. Xiao and M.Shah. "Motion Layer Extraction in the Presence of Occlusion using Graph Cut." In *Proc. IEEE CVPR*, pp. 972–979, 2004.

[193] J. Xiao and M.Shah. "Tri-view morphing." *Computer Vision and Image Understanding*, **96**(3), 2004.

[194] J. Xiao and M.Shah. "Accurate Motion Layer Segmentation and Matting." In *Proc. IEEE CVPR*, pp. 698–703, 2005.

[195] J. Xiao and M. Shah. "Two-Frame Wide Baseline Matching." In *Proc. IEEE ICCV*, pp. 603–609, 2003.

[196] A. Zisserman, P. Beardsley, and I. Reid. "Metric Calibration of a Stereo Rig." In *IEEE Wkshp on Representation of Visual Scenes*, pp. 93–100, 1995.

[197] B. Zitova and J. Flusser. "Image registration methods: a survey." *Image and Vision Computing*, **21**:977–1000, 2003.

[198] Z. Zhang. "Iterative point matching for registration of free-form curves and surfaces." *Int. J. Comput. Vision*, **13**(2):119–152, 1994.

[199] Z. Zhang. "Determining the Epipolar Geometry and its Uncertainty: A Review." *Int. J. Comput. Vision*, **27**(2):161–195, 1998.

[200] Z. Zhang. "A Flexible New Technique for Camera Calibration." *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**(11):1330–1334, 2000.

[201] Z. Zhang. "Camera Calibration with One-Dimensional Objects." In *Proc. ECCV*, pp. 161–174, 2002.

[202] Z. Zhang. "Camera Calibration with One-Dimensional Objects." *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(7):892–899, 2004.

[203] L. Zelnik-Manor and M. Irani. "Multi View Subspace Constraints on Homographies." In *Proc. IEEE ICCV*, pp. 710–715, 1999.

[204] H. Zhang, A. Kankanhalli, and S.W. Smoliar. "Automatic partitioning of full-motion video." *Multimedia Syst.*, **1**(1):10–28, 1993.

[205] C. Zitnick, S. Kang, M. Uyttendele, S. Winder, and R. Szeliski. "High-Quality Video View Interpolation Using a Layered Representation." In *Proc. ACM SIGGRAPH*, 2004.

[206] Z. Zhang and H. Tsui. "3D reconstruction from a single view of an object and its image in a plane mirror." In *Proc. ICPR*, volume 2, pp. 1174 – 1176, 1998.

[207] R. Zhang, P. Tsai, J. Cryer, and M. Shah. "Shape from Shading: A Survey." *IEEE Trans. Pattern Anal. Mach. Intell.*, **21**(8):690–706, 1999.

[208] D. Zongker, D. Werner, B. Curless, and D. Salesin. "Environment Matting and Compositing." In *Proc. ACM SIGGRAPH*, pp. 205–214, 1999.

[209] Y. Zhang, J. Xiao, and M. Shah. "Motion Layer Based Object Removal in Videos." In *Proc. WACV*, pp. 516–521, 2005.

[210] H. Zhang, G. Zhang, and K.-Y. K. Wong. "Camera Calibration with Spheres: Linear Approaches." In *Proc. IEEE ICIP*, volume II, pp. 1150–1153, 2005.